RL-TR-96-42
Final Technical Report
March 1996

A MODULATION MODEL FOR CHARACTERIZING SPEECH SIGNALS

Rutgers University

Dr. Richard J. Mammone, Dr. Khaled Assaleh,
Dr. Kevin Farrell, Dr. Ravi Ramachandran,
and Dr. Mihailo Zilovic

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Rome, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be releasable to the general public, including foreign nations.

RL-TR- 96-42 has been reviewed and is approved for publication.

APPROVED: Douglas G. Smith

DOUGLAS G. SMITH, 1Lt, USAF
Project Engineer

FOR THE COMMANDER:

DELBERT B. ATKINSON, Colonel, USAF
Director of Intelligence & Reconnaissance

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify Rome Laboratory (IRAA), Rome NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
March 1996

3. REPORT TYPE AND DATES COVERED
Final Sep 91 - Apr 95

4. TITLE AND SUBTITLE

A MODULATION MODEL FOR CHARACTERIZING SPEECH SIGNALS

5. FUNDING NUMBERS,

C - F30602-91-C-0120
PE - 62702F
PR - 4594
TA - I5
WU - I0

6. AUTHOR(S)

Dr. Richard J. Mammone, Dr. Khaled Assaleh,
Dr. Kevin Farrell, Dr. Ravi Ramachandran, and
Dr. Mihailo Zilovic

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Rutgers University, Busch Campus
CAIP Center
P.O. Box 1390
Piscataway NJ 08855-1390

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Rome Laboratory/IRAA
32 Hangar Rd
Rome NY 13441-4114

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

RL-TR-96-42

11. SUPPLEMENTARY NOTES

Rome Laboratory Project Engineer: Douglas G. Smith, 1Lt, USAF/IRRE/(315) 330-4024

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The purpose of this effort was to research and develop new techniques for modeling speech segments. These models were used for Speaker Identification and for Keyword Spotting. The Format Mode Modulation Model and Pitch Mode Modulation Model were investigated. A novel feature, the Advanced Cepstral Weighting (ACW) Cepstrum, was developed and implemented to enhance performance. These algorithms were implemented on workstations, and also on an nCUBE supercomputer. In the future, these research algorithms can be used as part of a larger application.

14. SUBJECT TERMS

Modulation model, Speaker identification, Keyword spotting

15. NUMBER OF PAGES
60

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

Contents

List of Figures

1 Introduction

1.1 Speaker Identification

Speech communication is the transfer of information from one person's mouth to another's ear using sound. Sound is variations in pressure which propagate as waves through the air medium and reach the listener's ear. The listener then deciphers these waves into a received message. Speaker recognition refers to the concept of recognizing a speaker by the sound of his/her voice. In automatic speaker recognition, an algorithm takes the listener's role in deciphering speech waves into either the underlying textual message or a hypothesis concerning the speaker's identity. When the task is to identify the person talking rather than what the person is saying, the speech signal must be processed to extract measures of speaker variability instead of segmental features. There are two methods for using speaker recognition technology, namely *speaker identification* (ID) and *speaker verification*. Some of the important applications of speaker recognition include customer verification for bank transactions, access to bank accounts through telephones, control on the use of credit cards, and for security purposes in the Army, Navy and the Air Force. Speaker recognition is described in detail in [60] [62].

Speaker identification deals with a situation where the person has to be identified from among a predetermined set of persons by using his/her voice samples. The objective of speaker verification is to verify the claimed identity of that speaker based on the voice samples of that speaker alone. For speaker recognition, the acoustic aspects of what characterizes the differences between voices are obscure and difficult to separate from signal aspects that reflect segment recognition. There are three sources of variation among speakers. They are

- Differences in vocal cords and vocal tract shape;
- Differences in speaking style; and
- Differences in what speakers choose to say [66].

Automatic speaker recognizers exploit only the first two variation sources, examining low-level acoustic features of speech, since a speaker's tendency to use certain words and syntactic structures (the third source) is difficult to quantify or control in an experiment. Most of the parameters and features used in speaker recognition problem contain information useful for the identification of both the speaker and the spoken message. The speaker ID problem may further be classified into *closed set* and *open set*. Closed set speaker ID problem refers to a case where the speaker is known *a priori* to belong to a set of M speakers, whereas in open set, the speaker may be out of the set. In open set problems, a scheme is used wherein a threshold value is needed in order to find out if the speaker is out of the set of M speakers.

1.1.1 Feature Extraction

Feature extraction is the process of deriving a compact set of parameters that are characteristic of a given signal. These parameters are desired to preserve all the information relevant to the application, and to have no redundancy in representing the signal. For speaker identification, a desired set of features is one that minimizes the intraspeaker variance and at the same time maximizes the interspeaker variances.

The majority of speaker identification systems use some type of short time spectral analysis followed by a certain transformation as a feature extraction step. Due to the short time stationarity of speech signals, short time spectral analysis is applied to overlapping segments (frames) of length 10 - 30 msec and and overlap of $\frac{1}{2}$ to $\frac{2}{3}$ the segment length. The short time spectrum is transformed into a sequence of feature vectors that compactly represents the underlying speech signal.

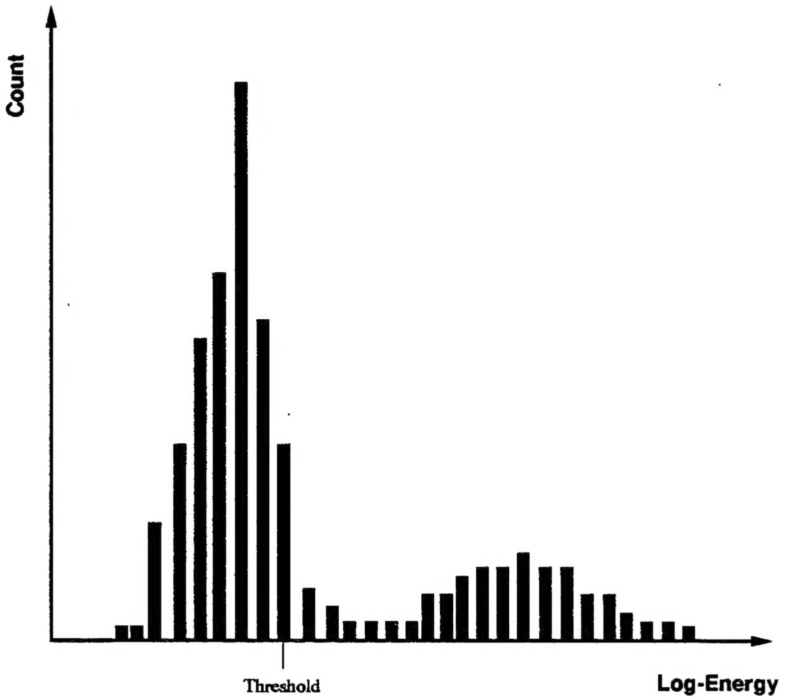


Figure 1: A typical histogram of the frame log-energies of conversational speech

The most effective and widely used spectral analysis techniques as front end processors for speech and speaker recognition applications are LP analysis and filter bank analysis. For the reasons listed below, this paper focuses on the LP-based front end processor.

Before performing any spectral analysis on the signal some preprocessing is required. Most important is speech/silence discrimination. The algorithm for silence removal is now described and then followed by descriptions of the different features that can be used by the system.

Silence Removal

Speech/silence discrimination is achieved by signal-dependent energy thresholding. For each utterance, the energy threshold is determined by constructing the histogram of the frame log-energies. Only frames with log-energies higher than the determined decided threshold are kept for further processing. The threshold is determined based on the fact that for spontaneous speech, an utterance typically contains a fair amount of silence or nonspeech. Therefore, the histogram of the frame log-energies shows a bimodal distribution as shown in figure 1. The distribution in the lower end of the log-energy axis corresponds to the silence, and the other distribution corresponds to the speech. The threshold point between silence and speech is chosen somewhere between the means of the distributions. The SNR of the signal is computed based on the determined threshold. Following the speech/silence discrimination step, the speech is processed by a single-tap high frequency preemphasis filter, and partitioned into 30 ms Hamming windowed overlapping frames at a rate of 100 frames/sec.

Linear Prediction Model for Feature Extraction

Besides being a good estimate to the source-filter speech production model [63], LP analysis [30, 31] gained its popularity for the following reasons:

- It is analytically and computationally tractable.

- It provides spectral estimates that are less biased and have lower variability than Fourier-based spectral estimates.
- In speech-pattern recognition applications, it has been found that LP-based front ends provide comparable or better performance than filter bank front ends [32]. Also, for speaker identification we have found in agreement with [33, 34] that LP based front ends perform at least as good as filter bank front ends.

The short-time transfer function of the LP model is given by:

$$H(z;m) = \frac{1}{A(z;m)} = \frac{1}{1+\sum_{i=1}^{P} a_i(m)z^{-i}} \tag{1}$$

where m is the frame index representing the time dimension, P is the order of the LP model, and $a_i(m)$ is the set of prediction coefficients of the m^{th} frame. $A(z;m)$ is the short-time LP polynomial.

Several sets of features can be derived from $H(z;m)$ [35, 59, 37].

Atal [35] provided a comparison of parameters obtained from linear prediction, the impulse response, autocorrelation, vocal tract area function, and cepstral coefficients and found the cepstrum to provide the best results for speaker recognition.

Another comparison between cepstrum and log area ratios (LARs) [59] for speaker verification concluded that cepstral coefficients outperform the LARs. For high quality speech, line spectral pairs (LSPs) are found to yield speaker identification rates that are comparable to or better than those of the cepstral coefficients [37]. However, for telephone quality speech, cepstral coefficients are found to be superior to LSPs. Today, cepstral coefficients are the dominant features used for speaker recognition [59, 48, 39].

In most practical applications, speech is collected under different environments and possibly through different communications channels. This causes a mismatch among corresponding reference and testing patterns. The characteristics of the cepstral coefficients have been extensively studied for the purpose of minimizing such a mismatch. In this regard, two major postprocessing steps have been introduced: intraframe processing known as cepstral weighting or liftering [32, 41, 42], and interframe processing which exploits the time evolution of the cepstral coefficients [45, 46, 47, 64]. The ACW scheme introduced in this paper falls within the intraframe processing techniques.

Due to the importance of the LP cepstral features in speech and speaker recognition, we dedicate a separate section to discuss their properties and their relations to other LP parameters.

LP Cepstrum

The short-time LP cepstrum is defined as the inverse z transform of the natural logarithm of the short-time LP transfer function $H(z;m)$. It can be viewed as the impulse response of $\ln H(z;m)$ which is given by:

$$\ln H(z;m) = \sum_{n=1}^{\infty} c_n(m)z^{-n} \tag{2}$$

where $c_n(m)$ is the n^{th} cepstral coefficient of the m^{th} frame.

A simple and unique recursive relationship between $c_n(m)$ and the prediction coefficients $a_n(m)$ can be obtained by differentiating both sides of the of (2) with respect to z^{-1} and equating the coefficients of equal powers of z^{-1}. This relation is given by [35]

$$\begin{aligned} c_1(m) &= -a_1(m), \\ c_n(m) &= -a_n(m) + \sum_{k=1}^{n-1}(\frac{k}{n}-1)a_k(m)c_{n-k}(m), \quad 1 < n \le P, \end{aligned}$$

$$c_n(m) = \sum_{k=1}^{n-1} (\frac{k}{n} - 1) a_k(m) c_{n-k}(m), \qquad n > P. \tag{3}$$

An alternative, rather more insightful, method of obtaining the short-time cepstral coefficients is by relating them to the poles of $H(z;m)$ and hence to the resonance center frequencies and bandwidths.

$$H(z;m) = \frac{1}{\prod_{i=1}^{P}(1 - z_i(m)z^{-1})} \tag{4}$$

By substituting (4) in (2) one gets

$$\sum_{i=1}^{P} \ln(1 - z_i(m)z^{-1}) = -\sum_{n=1}^{\infty} c_n(m) z^{-n}. \tag{5}$$

The factor $\ln(1 - z_i(m)z^{-1})$ can be expanded [65] as

$$\ln(1 - z_i(m)z^{-1}) = -\sum_{n=1}^{\infty} \frac{1}{n} z_i(m)^n z^{-n}. \tag{6}$$

By combining (5) and (6), $c_n(m)$ can be expressed in terms of the roots of the LP polynomial as follows.

$$c_n(m) = \frac{1}{n} \sum_{i=1}^{P} z_i(m)^n. \tag{7}$$

Thus $c_n(m)$ can be interpreted as the power sum of the LP polynomial roots normalized by the cepstral index [40].

Since $z_i(m)$ is associated with time varying center frequencies $\omega_i(m)$ and bandwidths $B_i(m)$ by the relation

$$z_i(m) = e^{-B_i(m) + j\omega_i(m)}, \tag{8}$$

the short time cepstral coefficients can be expressed as:

$$\begin{aligned} c_n(m) &= \frac{1}{n} \sum_{i=1}^{P} e^{-n(B_i(m) + j\omega_i(m))} \\ &= \frac{1}{n} \sum_{i=1}^{P} e^{-nB_i(m)} cos(n\omega_i(m)). \end{aligned} \tag{9}$$

Thus the n^{th} cepstral coefficient can be interpreted as a nonlinear transformation of the resonance center frequencies and bandwidths.

Robust Feature Processing

Cepstral Features are found to yield excellent performance for text-independent speaker identification when training and testing speech are collected under relatively high-quality stationary environments. However, in practical applications, the speech used by the system is subject to various sources of degradations such as background noise and communications channels variability. Such degradations often result in reduced recognition rates. This is due to the mismatch created among corresponding reference and testing patterns (in this case cepstral features $c_n(m)$). To minimize this mismatch, two major cepstral postprocessing approaches have been introduced: intraframe and interframe processing.

Intraframe Processing

Intraframe processing is also known as cepstral weighting or liftering. The rationale behind cepstral weighting is to account for the sensitivity of the low-order cepstral coefficients to the overall spectral slope and the sensitivity of the high-order cepstral coefficients to noise. In this regard several **fixed weighting** schemes have been recently introduced [43]. By "fixed weighting" we mean that the applied weights are only a function of the cepstral index n. Therefore these weights are fixed with respect to the frame index m. Generally, the resulting weighted cepstrum is given by

$$\tilde{c}_n(m) = w_n c_n(m), \tag{10}$$

where w_n is the cepstral weighting window (also known as the lifter).

The simplest and most straightforward weighting sequence is the rectangular weights, which have the effect of truncating the infinite cepstral sequence. Cepstral truncation has the effect of *slightly* smoothing the LP spectra. We say slightly because the LP cepstra are smooth by nature and hence need a relatively small number of parameters to determine them. Alternatively, the truncation is justifiable due to the fact that the LP cepstrum is the sum of exponentially decaying sequences that can be sufficiently represented by a finite number of terms L. Since for a P^{th} order all-pole spectrum the first P cepstral coefficients uniquely determine that spectrum, L is usually chosen to be equal to or greater than P. The advantages of cepstral truncations are:

- to reduce the dimensionality of the cepstrum so as to be usable as a feature vector, and
- to suppress the variability of the higher order cepstral coefficients.

Other more sophisticated weighting schemes that take advantage of the statistical characteristic of the cepstral coefficients have been recently introduced, These included bandpass liftering (BPL) [41] and quefrency liftering [42, 44].

BPL weights a cepstral sequence by a raised sine function of the form

$$w_n = \begin{cases} 1 + \frac{L}{2} sin\left(\frac{n\pi}{L}\right) & n = 1,\ 2,\ \ldots,L \\ 0 & otherwise \end{cases} \tag{11}$$

where L is normally chosen to be greater than P, the order of the LP model. The attenuation of the low-order coefficients is based on the fact that these coefficients are more susceptible to channel variations. The attenuation of the high-order coefficients is based on the same reason given for truncation.

Quefrency liftering applies an asymmetric triangular window of the form

$$w_n = \begin{cases} n & n = 1,\ 2,\ \ldots,L \\ 0 & otherwise \end{cases}; \tag{12}$$

where n is the cepstral index.

This weighting is based on the hypothesis that the standard deviations of the cepstral coefficients are inversely proportional to their cepstral index n. Thus, this weighting scheme approximates the statistical normalization approach which is accomplished by multiplying each vector of an array of vectors by the inverse of the covariance matrix of that array. Here the covariance matrix is assumed to be diagonal which is a valid assumption for the cepstral features. Other variations of quefrency liftering such as trapezoidal and symmetric triangular were also used [43]. These lifters were used to account for the sensitivity of high order cepstral coefficients to noise.

It should be noted here that fixed cepstral weighting can be incorporated in the distance measure between two unweighted vectors. For example, a weighted Euclidean distance measure between $c_n(i)$ and $c_n(j)$ is given by

$$d_{i,j} = \sum_{n=1}^{L} w(n)^2 (c_n(i) - c_n(j))^2 \tag{13}$$

The above mentioned fixed weighting schemes apply fixed weights to all the feature vectors extracted from an utterance assuming that all the frames undergo the same distortion. This assumption is not always applicable since in many practical cases distortions vary with time. In such cases an adaptive weighting scheme that is capable of adapting to the time-varying nature of the distortions is desired.

In the following we introduce a new adaptive weighting scheme which results in a new set of cepstral features that show robustness to channel variations.

Interframe Processing

Log Area Ratio The next feature investigated is the Log Area Ratio parameters. The Log Area Ratios are a bilinear transform of the reflection coefficients and are considered a very efficient transformation of the reflection coefficients for purposes of speech coding. They are related to the reflection coefficients by the formula:

$$g(l) = \frac{1}{2} \log \frac{1 + \kappa(l)}{1 - \kappa(l)} \qquad forl = 1, 2, ..., M \tag{14}$$

These log area ratios correspond to the cross sectional areas of the different sections of the vocal tract filter estimated by linear prediction.

ACW Cepstrum

The ACW scheme modifies the LP spectrum so as to emphasize the formant structure. This is achieved by operating on the different components of the spectrum, namely by amplifying the narrow-bandwidth components and attenuating the broad-bandwidth components. The resulting modified spectrum introduces zeros to the LP all-pole model. This is equivalent to a FIR filter that *normalizes* the contribution of the dominant modes of the signal (the formants).

For a given speech frame, the all-pole model can be expressed in a parallel form by partial fraction expansion:

$$H(z) = \frac{1}{1 + \sum_{i=1}^{P} a_i z^{-i}} = \sum_{i=1}^{P} \frac{r_i}{(1 - z_i z^{-1})}. \tag{15}$$

Since each pole z_i represents the center frequency ω_i and the bandwidth B_i of the i^{th} component, each component can be fully parameterized by ω_i, B_i, and expansion residue, r_i.

The sensitivity of the parameters (ω_i, B_i, r_i) with respect to channel variations have been experimentally evaluated by virtue of the the following experiment:

- A voiced frame of speech is processed through a random single-tap channel given by:

  $$\Theta_j(z) = 1 - a_j z^{-1} \tag{16}$$

  where a_j is a sequence of uniformly distributed random numbers between 0.0 and 1.0.

- The sequences of the parameters (ω_i, B_i, r_i) of all components are computed for each j in the random sequence a_j.

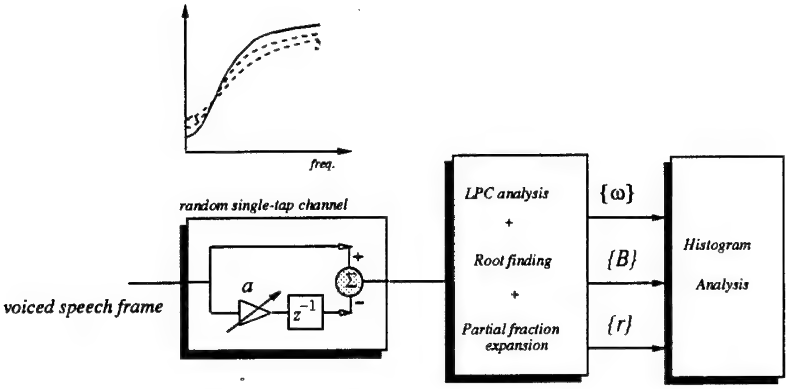


Figure 2: Block diagram of the experiment of the sensitivity of the LP spectral component parameters with respect to a random single-tap channel.

- Two sequences of (ω_i, B_i, r_i) are selected to represent a narrow-bandwidth component and a broad-bandwidth component.
- The sensitivity of the parameters of the selected narrow-bandwidth and broad-bandwidth components is evaluated by histogram analysis.

The block diagram of the experiment is shown in figure 2.

By examining the resulting histograms of the parameters of the broad-bandwidth component shown in figure (3), one concludes that the three parameters,(ω_i, B_i, r_i), associated with such components show large variances with respect to channel variations. Therefore, under channel variations, such components introduce undesired variability to the LP spectrum that results in a mismatch among testing and training patterns.

Narrow-bandwidth components tend to preserve their center frequencies and bandwidths since their histograms show small variances. However, the values of their residues demonstrate large variances. This effect is shown in figure (4).

These observations suggest guidelines to modify the LP spectrum so as to be robust for such variations. The modifications should be aimed at:

- eliminating the residues r_i from the LP spectrum, and
- attenuating the contribution of the broad-bandwidth components.

One way of achieving the suggested modifications is to normalize the residues, $\{r_i\}$, for example by setting $r_i = constant$, which can be viewed as weighting the i^{th} component by $\frac{1}{r_i}$. Normalizing $\{r_i\}$ results in a modified spectrum which we refer to as the ACW spectrum. The ACW spectrum is given by

$$\hat{H}(z) = \sum_{i=1}^{P} \frac{1}{(1 - z_i z^{-1})} = \frac{N(z)}{1 + \sum_{i=1}^{P} a_i z^{-1}}, \tag{17}$$

where

$$N(z) = \sum_{k=1}^{P} \prod_{i=1 \neq k}^{P} (1 - z_i z^{-1}), \tag{18}$$

which can be defactorized into the form

$$N(z) = P(1 + \sum_{i=1}^{P-1} b_i z^{-i}). \tag{19}$$

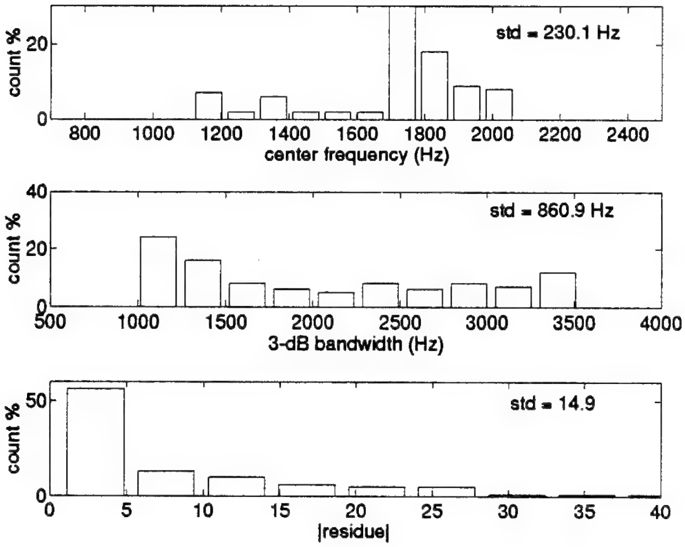


Figure 3: Histograms of the parameters of a broad-bandwidth component.

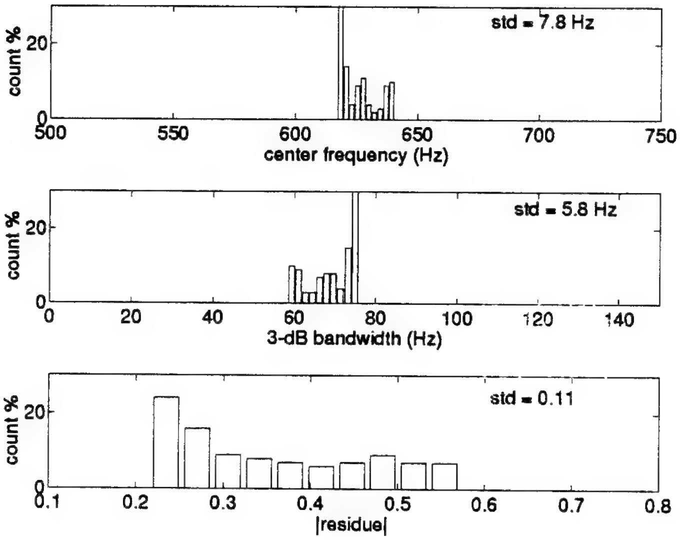


Figure 4: Histograms of the parameters of a narrow-bandwidth component.

By this modification to the LP spectrum the peak-value of each component is given by

$$\frac{1}{(1-z_i z^{-1})}|_{z=e^{j\omega_i}} = \frac{1}{1-|z_i|} \approx \frac{1}{B_i}. \tag{20}$$

Equation (20) shows that the ACW spectrum emphasizes the formant structure by weighting each component approximately by $\frac{1}{B_i}$. Thus narrow-bandwidth components are amplified and broad-bandwidth components are attenuated.

$\hat{H}(z)$ is no longer an all pole autoregressive (AR) transfer function, as it now has a MA filter represented by $P-1$ zeros. This MA filter introduced by normalizing the residues can be viewed as a FIR filter. This filter creates a spectrum whose components' peak values are inversely proportional to their bandwidths. This concept is illustrated in figure (5) where the components of the LP spectrum $H(z)$ and the ACW spectrum $\hat{H}(z)$ for a voiced speech frame are shown.

Figure (6) demonstrates the spectral mismatch created by a single-tap channel by showing the components of the LP spectrum of the same frame used in figure (5) after processing through $(1-0.9z^{-1})$.

In figure (7) the robustness of the ACW spectrum is demonstrated by showing that the same channel that disturbed the components of the LP spectrum has a very small effect on the components of the ACW spectrum.

The channel effect on the composite LP and ACW spectra is shown in figure (8). It is obvious that the mismatch between the LP spectra before and after processing through the channel is much larger than that between the corresponding ACW spectra.

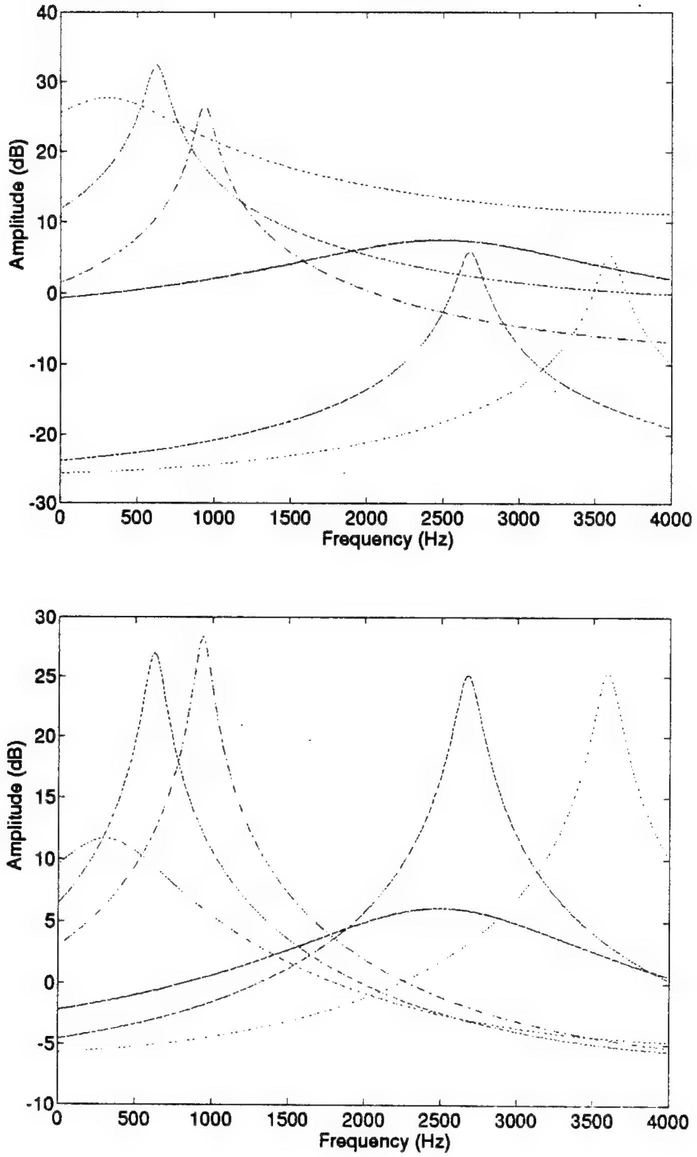


Figure 5: Components of (a) the LP spectrum, and (b) the ACW spectrum of a voiced speech frame

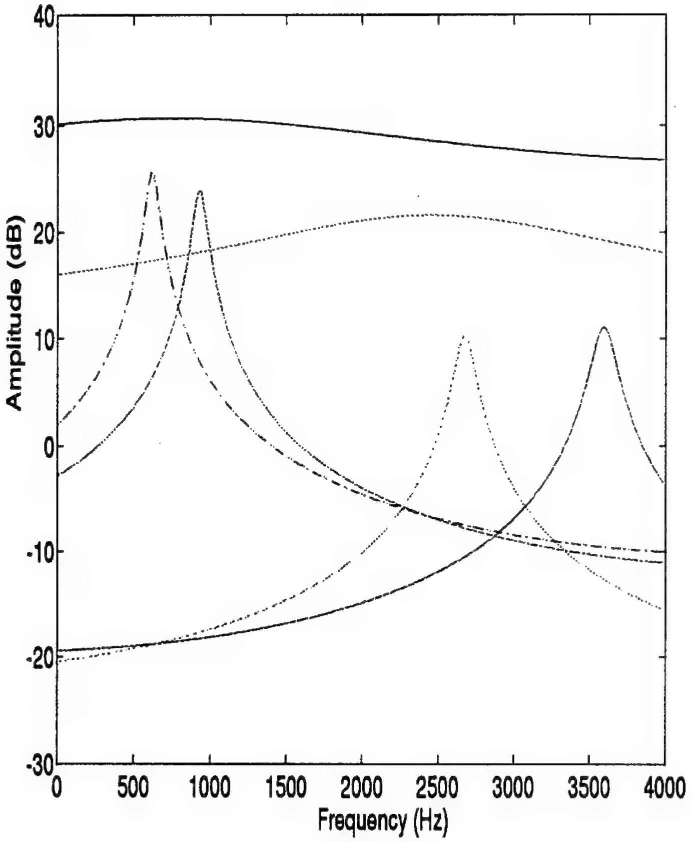


Figure 6: Components of the LP spectrum after processing through $(1 - 0.9z^{-1})$

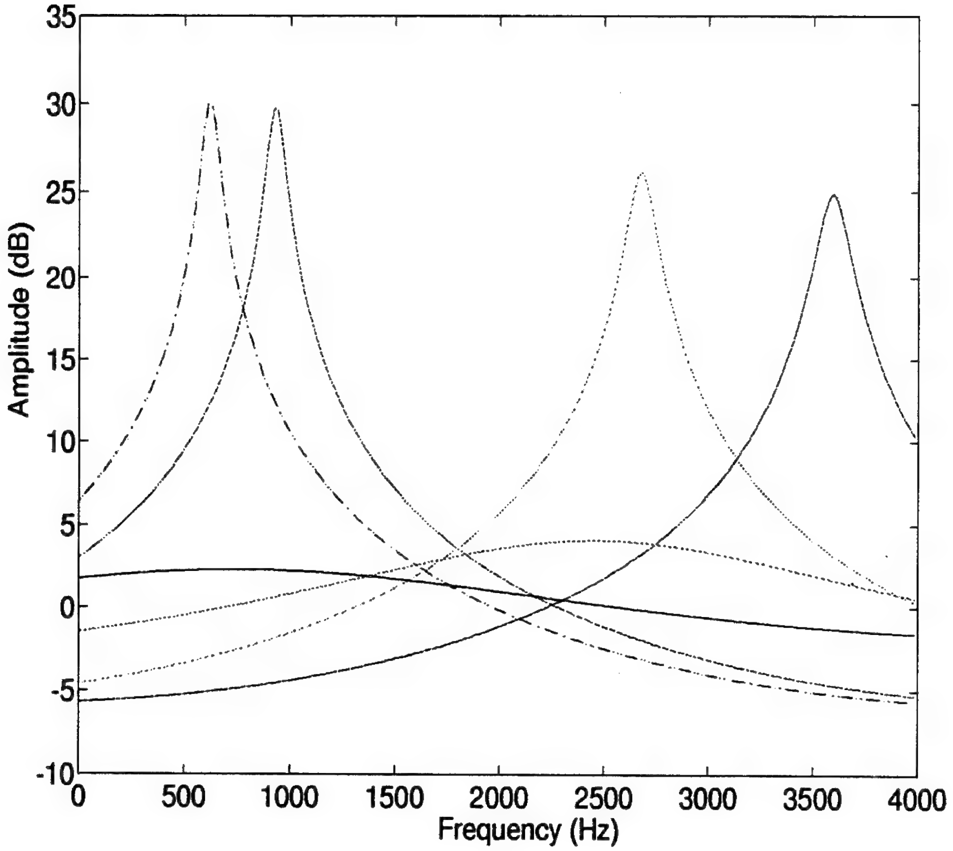


Figure 7: Components of the of the ACW spectrum after processing through $(1 - 0.9z^{-1})$

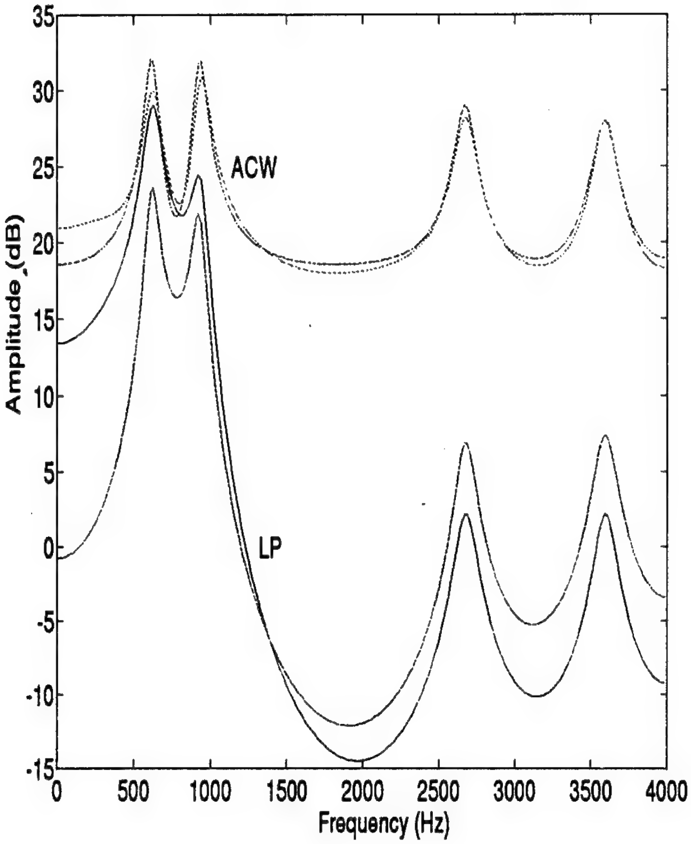


Figure 8: The channel effect on the composite LP and ACW spectra

2 Fast ACW cepstrum

Concept of ACW

A pth order LP analysis of speech leads to an LP polynomial $A(z)$ and an all-pole model $H(z) = 1/A(z)$ of the speech spectrum. The polynomial $A(z)$ is expressed as

$$A(z) = 1 - \sum_{i=1}^{p} a_i z^{-i} = \prod_{i=1}^{p}(1 - f_i z^{-1}) \tag{21}$$

which in turn can be guaranteed to be minimum phase by the autocorrelation method of LP analysis. The conventional LP cepstrum $c_{lp}(n)$ is defined for $n > 0$ and can be found by a recursion involving the coefficients a_i [32].

The approach in [67] is to first perform a partial fraction expansion of $H(z)$ to get

$$H(z) = \sum_{k=1}^{p} \frac{\lim_{z \to f_k} \left[(1 - f_k z^{-1})\,/A(z)\right]}{1 - f_k z^{-1}} = \sum_{k=1}^{p} \frac{r_k}{1 - f_k z^{-1}} \tag{22}$$

The experiments in [67] reveal that the residues r_k show considerable variations especially for nonformant poles when the speech is degraded by a channel. Therefore, the variations in r_k were removed by forcing r_k to be *constant* = 1 for every k. Hence, we get a pole-zero system function of the form

$$\frac{N(z)}{A(z)} = \sum_{k=1}^{p} \frac{1}{1 - f_k z^{-1}} \tag{23}$$

where

$$N(z) = \sum_{k=1}^{p} \prod_{i=1 \neq k}^{p} (1 - f_i z^{-1}) \tag{24}$$

which can be further written as

$$N(z) = p(1 - \sum_{k=1}^{p-1} b_k z^{-k}) \ . \tag{25}$$

Therefore, the ACW cepstrum is given by

$$c_{acw}(n) = c_{lp}(n) - c_{nn}(n) \tag{26}$$

for $n > 0$ where $c_{nn}(n)$ can be found by a recursion [32] involving the coefficients b_k.

It must be noted that the present method of finding $c_{acw}(n)$ from $A(z)$ involves the following steps [67].

1. Find $c_{lp}(n)$ from a_i
2. Determine the roots of $A(z)$
3. Find all the cofactors of $A(z)$ of order $p-1$ and add them up to get $N(z)$
4. Find $c_{nn}(n)$ from b_i
5. Find $c_{acw}(n) = c_{lp}(n) - c_{nn}(n)$

Steps 2 and 3 are mainly responsible for the increase in computational burden over merely finding $c_{lp}(n)$. As we shall see later, this increase is by a factor of 1.4. With the fast algorithm we propose, the increase in computation is a very small factor of 1.02.

2.1 Mathematical Definition of Numerator Polynomial

Theorem: Every single coefficient b_k of $N(z)$ in Eq. (25) is of the form

$$b_k = \frac{p-k}{p} a_k \tag{27}$$

$\forall k$, $1 \leq k \leq p-1$ where a_k is the kth coefficient of the LP polynomial $A(z)$ (see Eq. (21)).

Proof: Let

$$P_p(z) = \prod_{k=1}^{p} (z - z_k) \tag{28}$$

be a polynomial of order p with roots z_k. The derivative of $P_p(z)$ can be written as

$$\frac{d}{dz} P_p(z) = (z - z_i) \frac{d}{dz} \frac{P_p(z)}{z - z_i} + \frac{P_p(z)}{z - z_i} \tag{29}$$

$\forall i$, $1 \leq i \leq p$. The derivative term on the right hand side in the above equation is in turn a polynomial of order $p-1$ and therefore, can be further written as

$$\frac{d}{dz} \frac{P_p(z)}{z - z_i} = (z - z_j) \frac{d}{dz} \frac{P_p(z)}{(z - z_i)(z - z_j)} + \frac{P_p(z)}{(z - z_i)(z - z_j)} \tag{30}$$

$\forall j$, $1 \leq j \leq p \wedge j \neq i$. Therefore,

$$\frac{d}{dz} P_p(z) = (z - z_i)(z - z_j) \frac{d}{dz} \frac{P_p(z)}{(z - z_i)(z - z_j)} + \frac{P_p(z)}{z - z_i} + \frac{P_p(z)}{z - z_j} \tag{31}$$

Thus, by induction, the final expression for the derivative of $P_p(z)$ is obtained as

$$\frac{d}{dz} P_p(z) = \sum_{k=1}^{p} \frac{P_p(z)}{z - z_k} \tag{32}$$

By inspecting Eq. (32), it is obvious that the derivative of a polynomial of order p is equal to the sum of all the polynomial cofactors of order $p-1$. Since $N(z)$ is also a sum of all the LP polynomial cofactors of order $p-1$, the coefficients b_k in Eq. (25) are given as in the theorem.

2.2 Minimum Phase Property of Numerator Polynomial

In order to define the ACW cepstrum as in Eq. (26), it is necessary and sufficient that $N(z)$ be minimum phase. Here, we show that $N(z)$ is minimum phase.

Theorem: [68] Let $A \in C^{pxp}$. Then, each eigenvalue λ of the matrix A lies in one of the disks D_i in the complex plane

$$D_i = \{\lambda : |\lambda - \alpha_{ii}| \leq \sum_{j=1, j \neq i}^{p} |\alpha_{ij}|\} \tag{33}$$

$\forall i$, $1 \leq i \leq p$ where α_{ij} are the elements of the matrix A.

The set D_i are disks in the complex plane centered at α_{ii} and of radius $\sum_{j=1,j\neq i}^{p} |\alpha_{ij}|$. They are called Gerschgorin disks of the matrix A.

Suppose that the real matrix A has the companion form [69]:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & & & \vdots & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_p & -a_{p-1} & -a_{p-2} & -a_{p-3} & & -a_2 & -a_1 \end{bmatrix} \tag{34}$$

The characteristic polynomial of the matrix A is given as:

$$P_p^c(z) = z^p + \sum_{i=1}^{p} a_i z^{p-i} \tag{35}$$

Due to the companion form of the matrix A, there are only two different Gerschgorin disks

$$D_1 = \{\lambda : |\lambda| \leq 1\} \quad \textit{and} \quad D_2 = \{\lambda : |\lambda + a_1| \leq \sum_{j=2}^{p} |a_j|\} \tag{36}$$

containing the zeros of $P_p^c(z)$. Furthermore, if $P_p^c(z)$ is minimum phase, then $D_2 \subset D_1$ (see Fig. 1). The derivative of $P_p^c(z)$ is given as:

$$\frac{d}{dz} P_p^c(z) = pz^{p-1} + \sum_{i=1}^{p-1} (p-i) a_i z^{p-i} \tag{37}$$

Its roots are the same as the eigenvalues of the $(p-1) \times (p-1)$ matrix

$$A' = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & & & \vdots & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_{p-1}/p & -2a_{p-2}/p & -3a_{p-3}/p & -4a_{p-4}/p & & -(p-2)a_2/p & -(p-1)a_1/p \end{bmatrix} \tag{38}$$

Each eigenvalue of the matrix A' lies in one of the following two disks:

$$D_1 = \{\lambda : |\lambda| \leq 1\} \quad \textit{and} \quad D_3 = \{\lambda : |\lambda + (p-1)a_1/p| \leq \sum_{j=2}^{p-1} (p-j)|a_j|/p\} \tag{39}$$

Since

$$\sum_{j=2}^{p-1} (p-j)|a_j|/p < \sum_{j=2}^{p} |a_j| \quad \textit{and} \quad (p-1)|a_1|/p < |a_1| \tag{40}$$

disk D_3 is the shrinked version of the disk D_2 with its center translated along the real axis toward the origin. Since $P_p^c(z)$ is minimum phase, $D_2 \subset D_1$. Since D_3 is a shrinked and translated (toward the origin) version of D_2, it directly follows that $D_3 \subset D_1$ (see Fig. 1). Therefore, the minimum phase property of the derivative of $P_p^c(z)$ is established.

We have shown that if a polynomial is minimum phase, its derivative is also minimum phase. Hence, the minimum phase property of $A(z)$ ensures the minimum phase nature of $N(z)$.

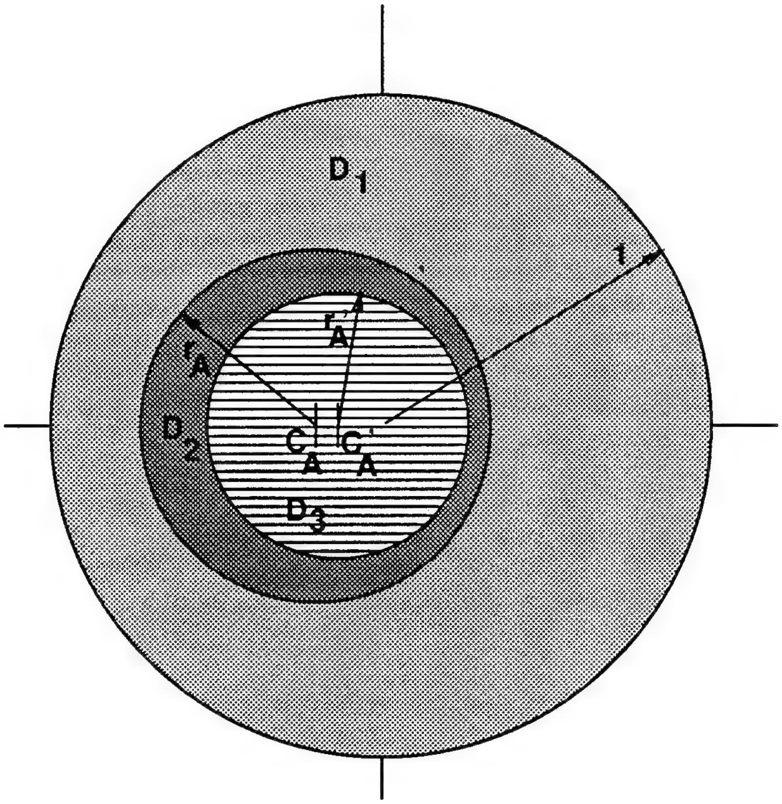


Figure 9: Gerschgorin disks D_1, D_2, D_3 $C_A = a_1$, $C'_A = (p-1)a_1/p$, $r_A = \sum_{j=2}^{p} |a_j|$, $r'_A = \sum_{j=2}^{p-1}(p-j)|a_j|/p$

2.3 Computer Time

Speech sampled at 8 kHz served as input to a system that does LP analysis and converts the LP coefficients to either the conventional cepstrum or the ACW cepstrum. An optimized software code that implements the above system was run on a SPARC10. Three different scenarios were compared in terms of CPU time. In scenario 1, the LP coefficients were transformed into $c_{lp}(n)$ via the well known recursion. In scenario 2, the LP coefficients were transformed into $c_{acw}(n)$ by the method offered in this paper for finding $N(z)$ and employing two separate recursions on $N(z)$ and $A(z)$. In scenario 3, the LP coefficients were again transformed into $c_{acw}(n)$ but unlike scenario 2, $N(z)$ was found by a standard polynomial root finding program [70]. The ratio of the required computer time for going from speech to cepstral features through scenarios 1, 2 and 3 is 1:1.02:1.40. This shows that our proposed method is much faster than doing polynomial root finding. Also, the more robust ACW cepstrum can be obtained by a negligible overhead as compared to the conventional LP cepstrum.

Frame Selection

As it is shown above, computing the component information (center frequencies and bandwidths) is an intermediate step in obtaining the ACW features. This information can be utilized as basis for selecting frames to be included in the sequence of feature vectors representing a given speech signal. This frame selection criterion is based on the following reasons.

- Voiced speech carry most of the speaker-dependent information.
- Speech frames of spectra that have apparent formant structure are the most discriminative and noise-robust frames.

Depending on the bandwidth of a given speech signal, one can devise a criterion for frame selection based on the formant information. This criterion can be summarized as follows. Frames that have certain number of resonances that lie within a specified frequency range, and have bandwidths smaller than a specified threshold are selected. This concept is depicted in figure (10).

Interframe Processing

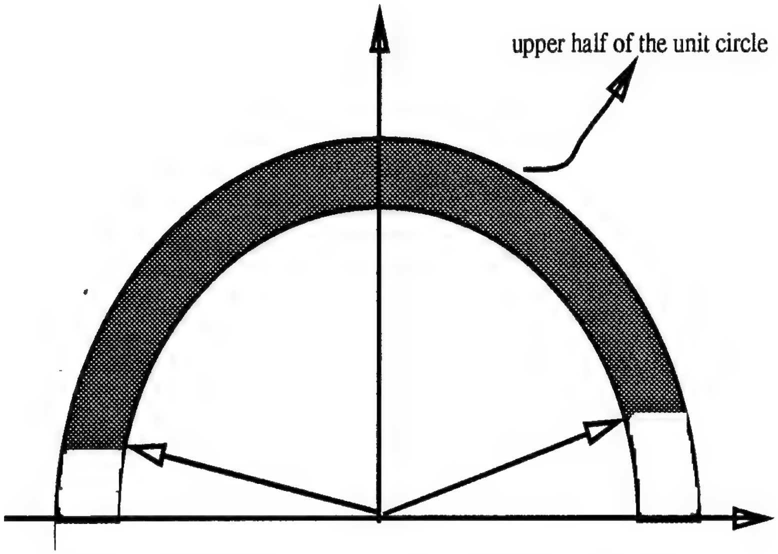
Unlike intraframe processing, interframe processing exploits the temporal variability of a sequence of feature vectors. The rationale behind interframe processing can be summarized by the following reasons:

- To emphasize the transitional information which is believed to provide orthogonal information to the instantaneous features obtained from the intraframe processing [45].
- To compensate for stationary and slowly varying linear channel effects that result in severe mismatch between training and testing data. This is achieved by removing time-invariant spectral information.

Transitional information is often referred to as dynamic features. In this paper we limit our discussion to interframe processing methods associated with spectral features in the cepstral domain.

It has been shown in [35] that the effect of any fixed frequency response distortion introduced by the recording apparatus or the transmission channel can be eliminated from a cepstral sequence simply by subtracting its long-term mean. It is interesting to notice that subtracting the long-term mean in the cepstral domain is equivalent to dividing by the geometric mean in the spectral domain:

$$c_n(k) - \frac{1}{M}\sum_{m=1}^{M} c_n(m) \Longleftrightarrow \frac{H(z;k)}{\prod_{m=1}^{M} H(z;m)^{\frac{1}{M}}}. \tag{41}$$



Frames that have a certain number of poles that lie within the shaded region are selected.

Figure 10: Frame selection based on formant information.

Spectral dynamic features are often represented by the time differential information of the cepstral sequence. The most straightforward representation being the first difference:

$$\Delta c_n(m) = c_n(m) - c_n(m-1). \tag{42}$$

However, first difference is susceptible to noise since it amplifies the high frequency components in the time trajectories of the cepstral coefficients. Therefore, the time derivative of $c_n(m)$ is approximated by polynomial approximation [59]. This approximation has the effect of bandpass filtering the time trajectories of $c_n(m)$ instead of the highpass filtering effect of the first difference. The filtered coefficients are known as the delta-cepstral coefficients and are given by:

$$\Delta c_n(m) = \sum_{k=-K}^{K} c_n(m-k)\delta(k), \tag{43}$$

where $\delta(k)$ represent the impulse response of the $2K+1$ taps bandpass filter which approximates the derivative of $c_n(m)$. The filter taps are given by:

$$\delta(k) = \frac{k}{\sum_{k=-K}^{K} k^2}, \qquad k = -K, \,, -K+1, \,\cdots, \, K \tag{44}$$

A typical value for K is 2 or 3. This technique is also known by $2K+1$ point regression-line.

Another intraframe processing technique is known as RASTA (RelAtive SpecTrA) [46]. Similar to the delta-cepstrum, RASTA has the effect of bandpass filtering the time trajectories of the cepstral coefficients. However, the RASTA filter includes a first order autoregression which has the effect of recursively removing the temporal average of the cepstral sequence. It also results in smoother cepstral trajectories due to the low-pass nature of the first order autoregression. The RASTA LP cepstrum $\Delta_R c_n(m)$ is given by

$$\Delta_R c_n(m) = \sum_{k=-K}^{K} c_n(m-k)\delta(k) + \alpha \Delta_R c_n(m-1) \tag{45}$$

where α is the coefficient of the first order autoregressive filter. This coefficient has a typical value of 0.98 [46]. To show the effect of intraframe processing on the short-time cepstral trajectories in the frequency domain, the frequency responses of the first difference, delta-cepstrum, and the RASTA filter are shown in figure (11). Notice that the frame sampling frequency is 100 frames/sec.

2.3.1 Speaker Modeling

Neural Tree Network

The supervised classifier considered here is the modified neural tree network (MNTN) [49]. The NTN [56] is a hierarchical classifier that combines the properties of decision trees [57] and feed-forward neural networks [58]. Whereas the NTN is strictly a classification tree, i.e., only the leaf labels are used, the MNTN additionally uses probability measures at the leaf nodes.

The assignment of probability measures occurs within a technique called *forward* pruning. The forward pruning algorithm consists of simply truncating the growth of the tree beyond a certain level. For the leaves at the truncated level, a vote is taken and the leaf is assigned the label of the majority. In addition to a label, the leaf is also assigned a confidence. The confidence is computed as the ratio of the number of elements for the vote winner to the total number of elements. The confidence provides a measure of confusion for the different regions of feature space. The concept of forward pruning is illustrated in Figure 12.

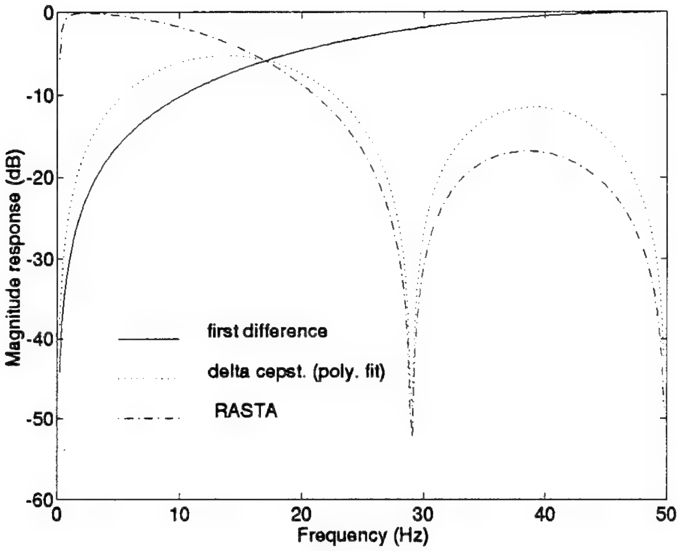


Figure 11: Frequency responses of various interframe filters

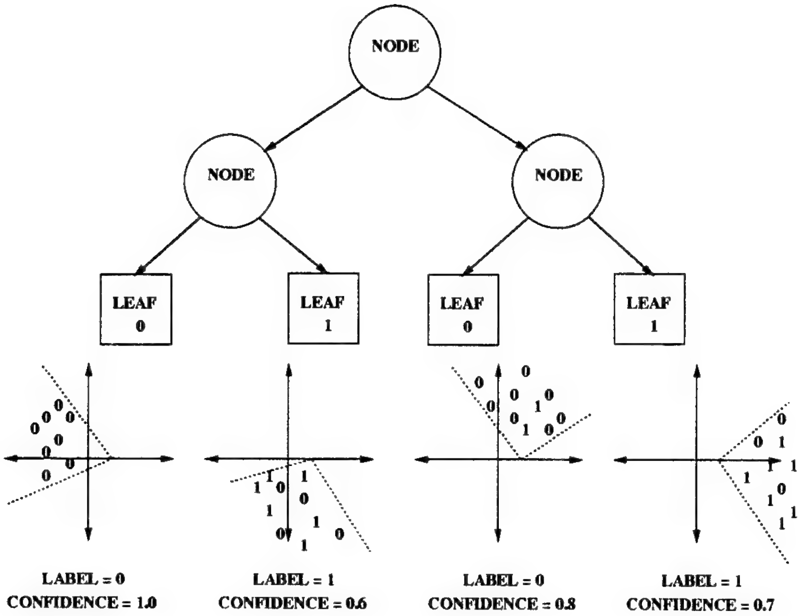


Figure 12: Forward Pruning and Confidence Measures

A MNTN can be trained for each speaker in the population as follows. First, the MNTN for each speaker is presented with a training set that is comprised of the data for all speakers. Here, the extracted feature vectors for that speaker are labeled as "one" and the extracted feature vectors for everyone else are labeled as "zero". A binary MNTN for speaker i is then trained with this data. This procedure is repeated for all speakers in the population.

Specifically, a trained MNTN can be applied to speaker recognition as follows. Given a sequence of feature vectors x from the test utterance and a trained MNTN for speaker S_i, the corresponding speaker score is found as:

$$P_{MNTN}(x|S_i) = \frac{\sum_{j=1}^{M} c_j^1}{\sum_{j=1}^{N} c_j^0 + \sum_{j=1}^{M} c_j^1}, \tag{46}$$

where c^1 and c^0 are the confidence scores for the speaker and antispeaker, respectively. Here, the M and N correspond to the number of vectors classified as the "one" and "zero", respectively.

Vector Quantization

The unsupervised classifier considered here is vector quantization (VQ). The VQ algorithm is based on clustering. This falls under the category of unsupervised training, i.e., the class label is not used. Clustering will automatically group the training data into its individual modes or classes. Numerous VQ algorithms exist, including the Linde-Buzo-Gray (LBG) [54] method and K-means algorithm [55]. The LBG method is used here.

The VQ classifier can be used for speaker recognition [48] as follows. Given the extracted feature vectors from a speaker, a codebook is constructed for that speaker. This process is repeated for all speakers in the population. For speaker identification, the feature vectors from a test utterance are applied to each of the codebooks. For a given codebook, the centroid that is closest to the test vector is found and the distance to this centroid is accumulated for that codebook. This process is repeated for all test vectors and the speaker is selected as corresponding to the codebook with the minimum accumulated distance.

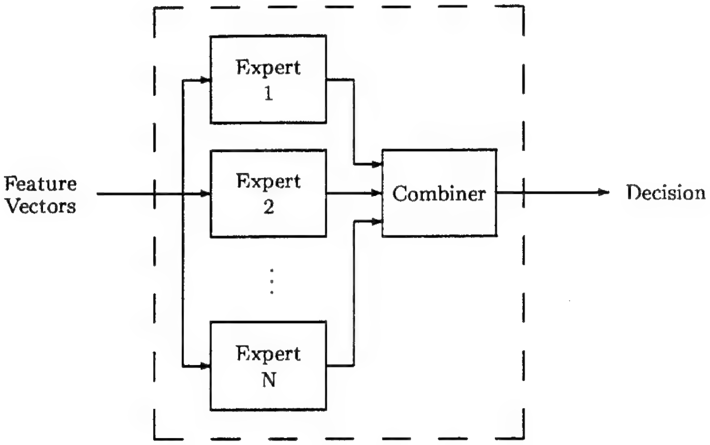


Figure 13: Data Fusion System

3 Data Fusion

It is often advantageous to combine the opinions of several experts when making a decision. For example, when one is obtaining a medical diagnosis, a decision for subsequent care may become easier after obtaining several opinions as opposed to just one. This concept has been exploited in the field of data fusion for tasks including handwriting recognition [50], remote sensing [52], etc.

The general form of a data fusion system is illustrated in Figure 13. Given a set of feature vectors, each expert outputs its own observation, which can consist of a probability measure, class label, etc. The combiner will then use one of many methods to collapse these observations into a single decision. The set of feature vectors can also be different from classifier to classifier, which would be a case of *sensor* fusion [51]. However, the work in this chapter only considers the.case for different experts and not different features.

There are numerous ways to combine the opinions of multiple experts. For example, if the outputs of all experts are probabilities then a simple combination method would be to take a weighted sum of the probabilities or of the logs of the probabilities. These methods are known as the linear opinion pool and log opinion pools [52]. If the outputs of the experts are class labels, then methods such as voting [50] or ranking [53] can be used. For fuzzy decisions, Dempster-Shafer theory can also be used for the combination of experts [50]. This chapter evaluates the linear and log opinion pool methods for speaker recognition. These methods are described in more detail as follows.

3.1 Linear Opinion Pool

The linear opinion pool is a commonly used data fusion technique that is convenient due to its simplicity. The linear opinion pool is evaluated as a weighted sum of the classifier outputs:

$$P_{linear}(x) = \sum_{i=1}^{n} \alpha_i p_i(x), \tag{47}$$

where $P_{linear}(x)$ is the probability of the combined system, α_i are weights, $p_i(x)$ is the probability of the individual classifier, and n is the number of classifiers. For all experiments in this paper, α is between zero and one and the sum of the α's is equal to one.

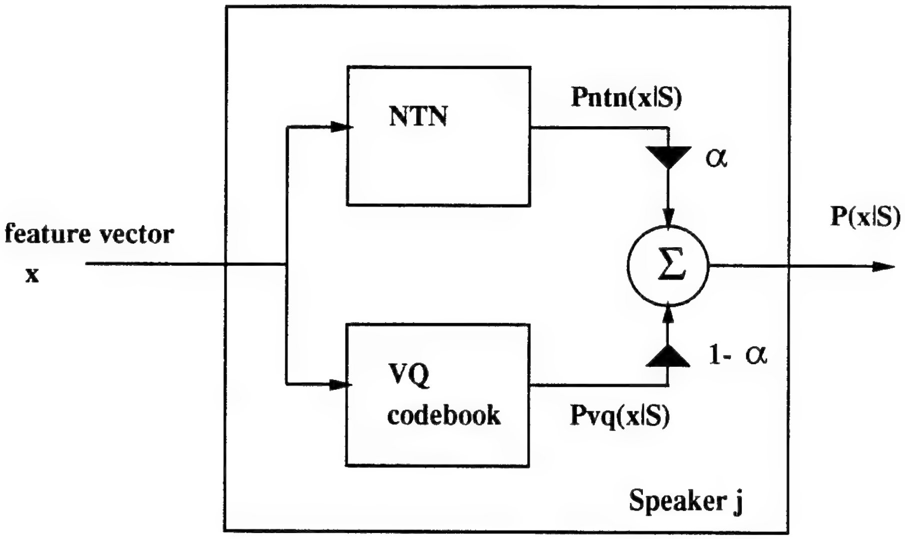


Figure 14: Data Fusion Approach

The linear opinion pool is appealing in that the output is a probability distribution and the weights α_i provide a rough measure of the expertise of the i^{th} expert. However, it is noted that the probability distribution of the combiner may be multimodal, which may impose a more complicated decision strategy. The linear opinion pool has been considered in speaker recognition for the combination of features [45], namely cepstrum and delta cepstrum features.

3.2 Log Opinion Pool

An alternative to the linear opinion pool is the log opinion pool. If the α weights are constrained to lie between zero and one and sum up to one, then the log opinion pool also outputs a probability distribution. However, as opposed to the linear opinion pool, the output distribution of the log opinion pool is unimodal [52].

The log opinion pool consists of a weighted product of the classifier outputs:

$$P_{log}(x) \;=\; \prod_{i=1}^{n} p_i^{\alpha_i}(x). \qquad (48)$$

Note that with this formulation, if any expert assigns a probability of zero, then the combined probability will also be zero. Hence, an individual expert has the capability of a "veto", whereas in the linear opinion pool the zero probability would be averaged in with the other probabilities.

One problem that both the linear and log opinion pools are subject to is the selection of the weights α_i. Several heuristic solutions [52] to this are to 1) use equal weights, i.e., $\alpha_i = 1/n$, 2) use weights proportional to a ranking, i.e., $\alpha = r/\sum_{r=1}^{n} r$, or 3) evaluate the weights over the range of zero to one for cross-validation data and select the best α_i.

3.3 Text-Independent Speaker Identification Using Data Fusion

Data fusion principles are used to combine the outputs of the NTN and VQ classifiers. The method used here is the linear opinion pool. This consists of evaluating a weighted sum of the classifier outputs as illustrated in Figure 14.

Here, the outputs of the NTN and VQ classifiers are multiplied by α and $1-\alpha$, respectively, where α lies between zero and one. Hence, when $\alpha = 0$ the system consists of solely the VQ classifier and likewise when $\alpha = 1$, only the NTN is used.

To enable the VQ and NTN classifiers to be used in such a system, several normalization steps must be used. First, the VQ distortion and NTN confidence must be converted to the same scale. Here, the VQ distortion and NTN confidence are normalized to lie on a scale of 0 to 1, where 1 denotes a perfect match. These scaled scores are analogous, though not equivalent, to probabilities.

The VQ distortion is normalized as follows:

$$P_{vq}(x|S_j) \;=\; e^{-(x-c_i)^2}, \tag{49}$$

where c_i is the centroid closest to x. The NTN label and confidence, which lies between 0.5 and 1.0, can be normalized to a single score as follows:

$$P_{ntn}(x|S_j) \;=\; \begin{cases} 0.5 * (1.0 + confidence), & if\ label = 1 \\ 0.5 * (1.0 - confidence), & if\ label = 0 \end{cases} . \tag{50}$$

These scores can now be combined and evaluated for speaker recognition tasks.

3.3.1 Confidence Measures

The overall confidence measure is calculated as a weighted linear combination of three individual measures. The first measure is based on the mismatch in the SNR of the training and testing data. The second measure is based on the channel mismatch between the training and testing data. The third measure is based on the amount of training and testing time.

SNR Mismatch

Any SNR mismatch between the training and testing data results in a degradation in the confidence with which a decision can be based. A confidence level is computed based on the determinations of the SNR of the training and testing data. The absolute value of the difference in the SNR values is found as the SNR mismatch. The confidence level is computed from this mismatch.

Speaker identification experiments for various SNR mismatches were conducted to get an identification success rate. From a discrete set of points relating the success rate to the SNR mismatch, a continuous functional fit is obtained. This function is specified as one of two fourth order polynomials depending on the SNR mismatch. During actual operation, the function value is calculated after finding the SNR mismatch. This value is the expected success rate or equivalently the confidence level. The memory requirement consists of only the polynomial coefficients.

Channel Mismatch

A channel mismatch between the training and testing data results in a degradation in the confidence with which a decision can be based. A confidence level is computed based on a quantitative determination of the channel mismatch between the training and testing data. Let $\mathbf{c}_{tr}$ ne the cepstral mean vector for the training data. Similarly, let $\mathbf{c}_{tt}$ be the cepstral mean vector for the test data. The channel mismatch is

$$20 \log \frac{||\mathbf{c}_{tr}||_2}{||\mathbf{c}_{tr} - \mathbf{c}_{tt}||_2} \tag{51}$$

The confidence level is computed from this mismatch.

Speaker identification experiments for various channel mismatches were conducted to get an identification success rate. From a discrete set of points relating the success rate to the channel mismatch,

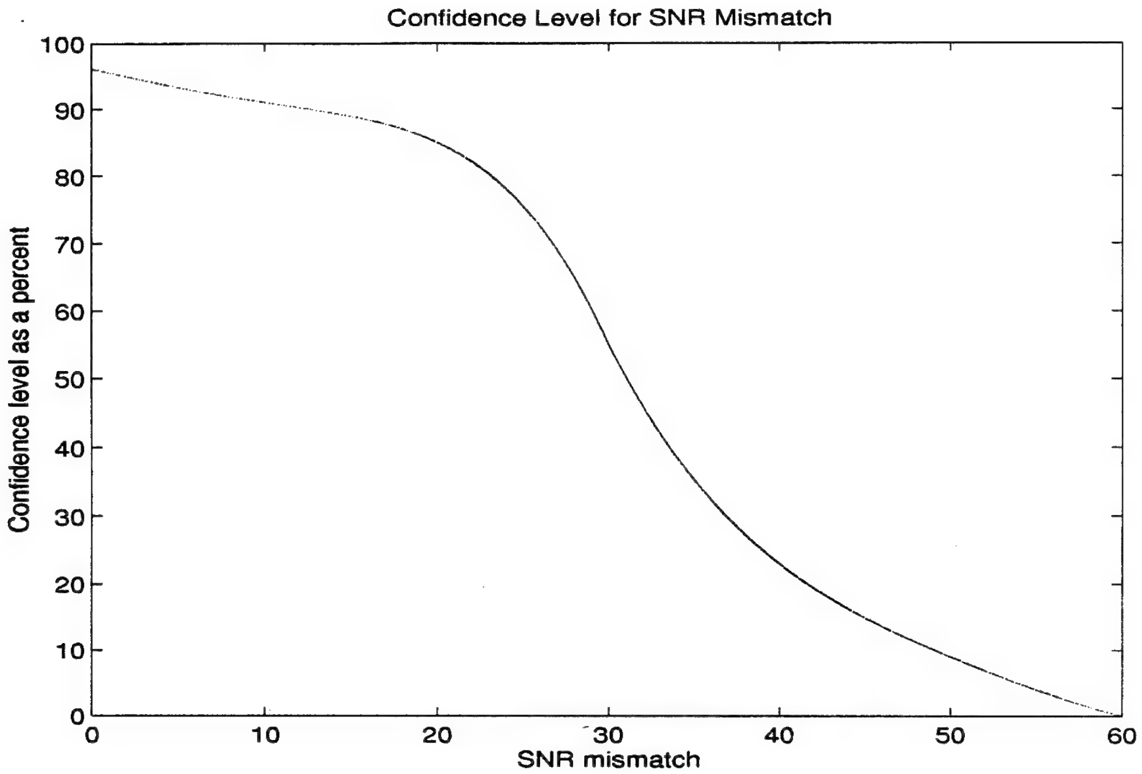


Figure 15: SNR confidence

a polynomial fit is obtained. During actual operation, the polynomial value is calculated after finding the channel mismatch. This value is the expected success rate or equivalently the confidence level. The memory requirement consists of only the polynomial coefficients.

Training/Testing Time

For a given training time of 3 seconds, the identification success rate is found for testing times up to 3 seconds. From this, a functional fit relating the success rate or the confidence level to the testing time for a given training time is obtained. A family of functions based on this fit is derived for various training times. Therefore, knowledge of the training and testing times will yield the confidence level by evaluation of one of the members of the family of functions. When the testing time exceeds 3 seconds, it is assumed to be 3 seconds for the purposes of finding the confidence. With regard to memory, it suffices to store the function originally obtained for a training time of 3 seconds.

4 Word Spotting

Word spotting is the process of locating a predefined keyword utterance within a continuous speech utterance. Word spotting is a subset of the general speech recognition task consisting of a limited vocabulary of keywords and a method of detecting words not in the vocabulary. Training and modeling of word spotting systems can fall under two categories, speaker dependent and speaker independent. Speaker dependent systems recognize spoken words from a specific speaker. Speaker independent systems are trained from speakers from a sample population and are expected to generalize to other populations.

Early work in the area of word recognition concentrated on methods based on [2][3][4][13][24][6] elastic template matching schemes using extracted speech features with a dynamic programming algorithm [1].

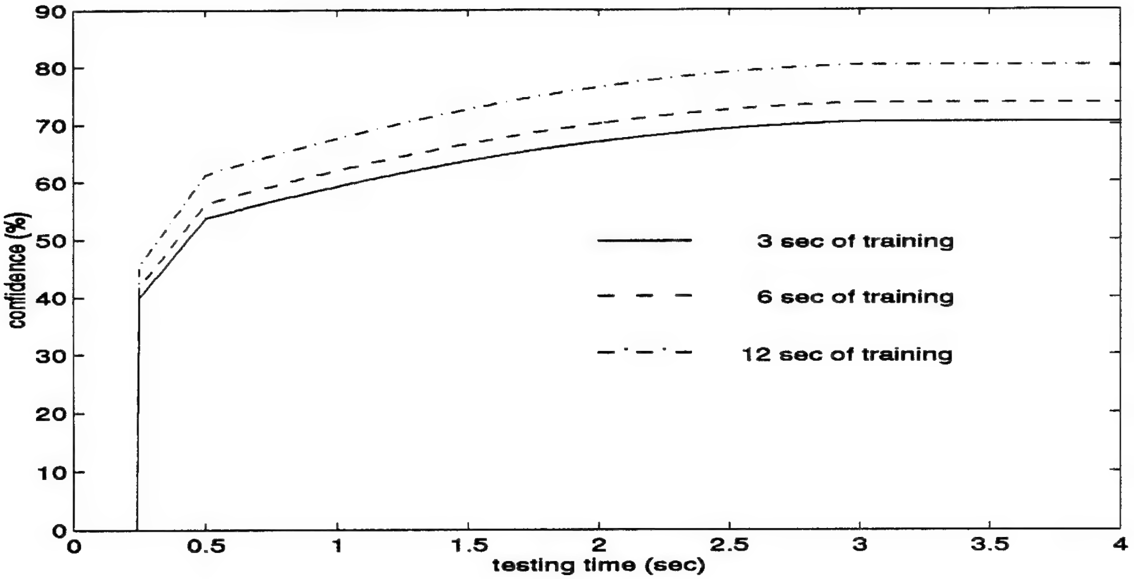


Figure 16: Training/testing time confidence

Some of these systems avoided the problem of detecting the word boundaries by either using a separate boundary detection scheme or requiring the utterance to contain only a single word. A template matching scheme transforms a segment of a speech utterance into a multidimensional vector. These temporally aligned vectors are used to construct a reference template. The reference template is then used to classify unknown speech to the nearest template based on a distance metric. Related systems using Hidden Markov Models for isolated word recognition grew out of the research in template matching systems [14][15][27].

Word spotting from continuous unconstrained speech allows a more flexible user interface, transferring the burden of word boundary detection from the user to the machine for "hands-free" machine control applications. Other applications of continuous word spotting systems include monitoring keywords within transmitted radio communications, voice activated calling and telephone routing systems. Early studies on automatic operator assisted telephone applications showed that approximately 20% of the callers, when asked to give an isolated word response, added extraneous speech within the response [24][26][27]. Clearly a robust automated system must include methods to disregard extraneous speech.

Many of the these speech recognition systems avoid practical issues, such as real-time or fast response of the system to a spoken word. The approach taken by a few systems focused on a low level parallelism of the underlying recognition system [6][23], but many of the practical issues are neglected for constructing a real-time recognition algorithm. Some work on using only a partial Viterbi backtrace during the recognition process in an HMM system allows faster performance without excessive loss of accuracy [18][22], but the algorithm still requires excess speech lag time for the backtrace, hindering a real-time response.

4.1 Neural Network Based Systems

Approaches using neural networks have also been suggested for word and phoneme recognition applications. Incorporation of neural networks into speech recognition systems allows the use of discriminative training to enhance the recognition performance. These systems range from those entirely based on neural network technology to hybrid approaches. Some of these systems use neural networks directly for subword recognition [21][19][29], or the construction of whole word models [10]. Hybrid approaches combine multiple neural networks or other classifiers systems in a hierarchical post processing approach [11][12][28] or a data fusion approach [8][20], combining the outputs of multiple classifiers.

4.2 Replacing Gaussian Mixture Models in an HMM with CDNTN Models

The context dependent models improves discrimination between subword models. By combining the NTN and the Gaussian mixture model presently used in the HMM to model the output probabilities, the discrimination ability of the NTN will improve the performance. The new CDNTN model provides one example of combining the two models. By using the CDNTN to model the posterior probability of a subword segment within a keyword, the subword models can be connected together to form a Markov chain. For word spotting applications the discriminate training data of the CDNTN model can be constructed to allow better discrimination between subword models within a keyword. One method is to apply discriminate training between subword state models. Feature vectors common to one subword difficult to separate from vectors assigned to other subwords, will be grouped together by the NTN in a region with low confidence or low posterior probability. Feature vectors that are well separated from those assigned to other subwords, will be grouped by the NTN into high confidence regions. This can allow a natural partitioning of the subword data so that complex distributions can be more accurately modeled.

Once a Markov chain is created for each keyword, state durations can be extracted from the training data and clustered to form a state duration template to non-parametrically model the durations of each state. For testing, a dynamic time warping algorithm can be used to evaluate the state outputs for the test utterance against the state duration model extracted from the training data. The state duration template provides a temporal model for the state outputs during a keyword occurrence distinguishing between random state outputs and temporally aligned outputs during a keyword occurrence obviating the need for a recognition network.

4.2.1 Feature Extraction

Mel-warped Cepstrum

Spectral analysis using a bank of filters is a well known front-end processor for speech recognition systems. Generally the filter bank is non-uniform and the spacing of the filters is based on critical bands proposed by perceptual studies. The critical band is almost linear for frequencies below $1000Hz$, and is almost logarithmic for frequencies above $1000Hz$. Mel scale is an approximation to the critical band scale. The relationship between frequency f $(inkHz)$ and the mel scale is approximated by the following equation

$$mel = 1000 \log_2(1 + f) \tag{52}$$

Thus, the mel-scale filter bank has filters spaced uniformly on a mel-frequency scale. Generally, the individual filters have a triangular bandpass frequency response, and the spacing between the filters as well as their bandwidths are determined by a constant mel frequency interval. An overlap equal to half the bandwidth is typically present between adjacent fiters. A mel-scale filter bank is illustrated in figure 17

Suppose a mel-scale filter bank of Q filters is designed, and the magnitude outputs within each filter is m_i $(i = 1, \cdots, Q)$. These magnitude outputs m_i are then converted into mel-frequency cepstral coefficients (MFCC) c_j $(j = 1, \cdots, M)$ by applying the discrete cosine transform as follows

$$c_j = \sum_{i=1}^{Q} m_i \cos(\frac{\pi i}{Q}(j - 0.5)) \quad (j = 1, \cdots, M) \tag{53}$$

where, M is the cepstral order.

4.3 Word Modeling

CDNTN

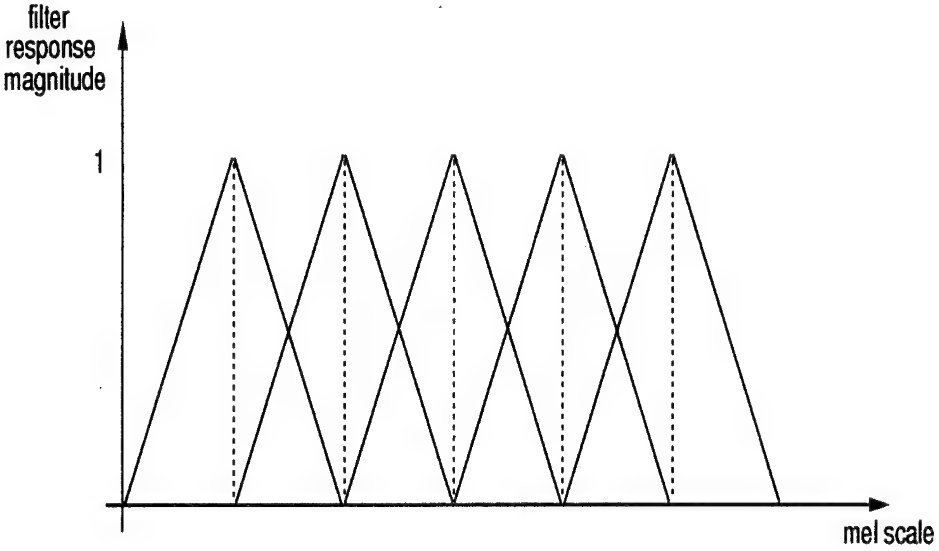


Figure 17: A mel-scale filter bank, with constant mel frequency bandwidth and overlap equal to half the bandwidth

4.3.1 CDNTN Subword State Model Description

The CDNTN modeling mechanism is formed by blending the NTN with a mixture model used in an HMM. This can be described in HMM terms as constructing multiple parallel states for each subword model. This is depicted in Figure 18 by considering that each feature vector presented to the subword model is first passed through an NTN network, which determines which parametric model will be used to generate the output probability for the subword. This method allows parallel parametric models to be naturally formed for multiple vocalizations of the same subword from different speakers. Each vocalization is then weighted by the confidence of that sound as a representative example from the training data base. The separation of the parallel parametric models is accomplished by the NTN by minimization of the cost function used to grow the tree. For the application in the word spotting system, this cost function chooses the most likely subword sound with respect to all the possible subword sounds. Those nearest in feature space are grouped together by the NTN hyperplanes. Once the NTN decides which substate model to use, the parametric model, weighted by the confidence of that region of feature space, is chosen by the multiplexer as the subword probability.

4.3.2 Modeling HMM State Durations using Dynamic Time Warping

As an alternative to modeling state durations and state transitions within an HMM model by transition probabilities, some HMM systems have been reported which replace the transition probabilities with explicit state duration models [5] [16] [17]. These state duration models are usually built into the HMM network as a parameterized function of time, and relax the constraints of the Markov property of the probability of state occupation. As an alternative to parametric modeling of the state durations, a non parametric method was developed by explicitly modeling the state occupations, and creating a template model which uses dynamic time warping (DTW) to warp the speech to account for time rate variations.

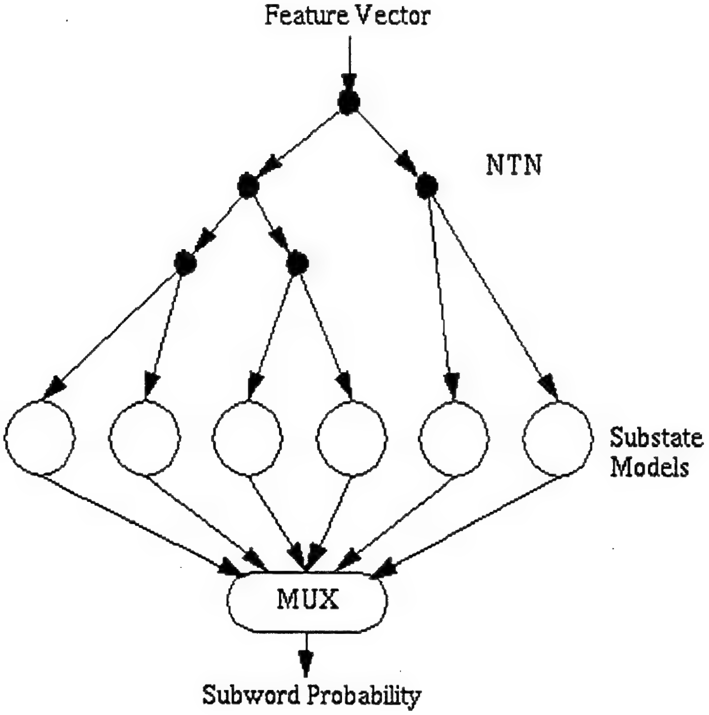


Figure 18: CDNTN Subword State Model

4.3.3 Dynamic Time Warping

Dynamic time warping is the process of applying a more general dynamic programming algorithm to optimally time align a sequence of vectors to a reference sequence. Dynamic programming was first described by Bellman [1] in 1957 as a method for the solution of multi-stage decision processes. These decision processes were described by a sequence of physical systems where each stage was characterized by a small set of state variables or parameters. At each stage there is a choice of a number of decisions. An assumption is made that the solution of the present state is independent of the past solutions. The dynamic programming algorithm maximizes some function on the state variables by making locally optimum decisions. As a discrete example, each state of the system can be described by a vector whose components vary with each discrete time step. Dynamic time warping applies the principles of dynamic programming to non linearly warp the time axis of sequence of vectors to optimally match a reference set. This warping process has the effect of stretching or compressing the time sequence of vectors to provide the best match to a reference template.

Given a multi-dimensional feature vector that varies with time, a template is created by ordering each reference template in an array for a fixed duration. The template duration or length is typically the average duration of a keyword. Typically the template is comprised of a set of multi-dimensional cepstral coefficients derived within a fixed time frame. This template is used with a test frame of unknown speech features to create a search grid for the DTW search depicted in Figure 19. In Figure 19 the j axis is defined as time samples for the reference template and the i axis as the time samples for the test pattern. The reference pattern of length J is placed against the j axis and the test pattern of length I is placed along the i axis. The DTW algorithm finds the locally optimal match in time between the reference template and test pattern. At each grid point along a row defined by the reference template, the distance between the

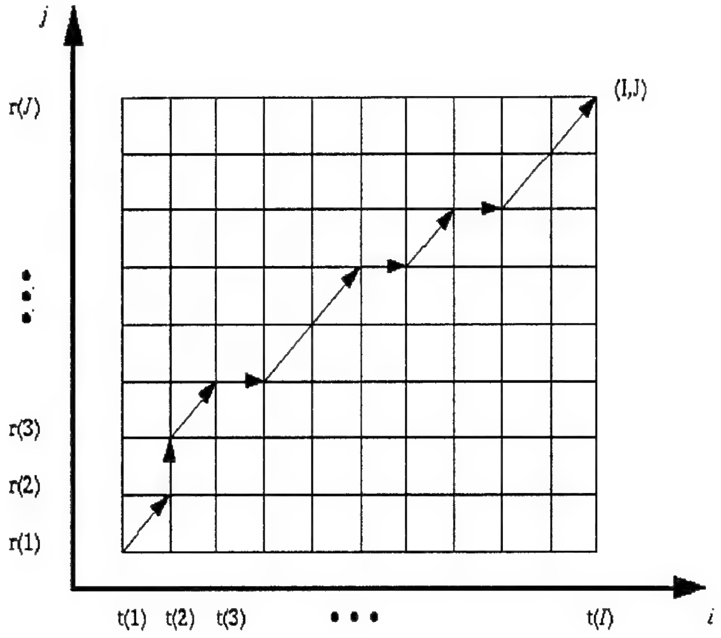


Figure 19: Test and reference patterns for DTW template matching

two templates are calculated. The grid point corresponding to the nearest distance between the template samples is selected as the locally optimum match, and the algorithm is repeated until the final match point (I,J) is reached. At each match point, the distance is accumulated and when the final match point is reached the accumulated distance score is available. The optimal search path is represented in Figure 19 by the bold lines through the grid. Many variations and constraints have been developed [5] to constrain the search within reasonable limits to avoid unnatural jumps in time.

4.3.4 Non Parametric State Duration Models Using Dynamic Time Warping

The use of dynamic time warping to find the optimal sequence of state outputs in a Markov chain of states is similar to a Viterbi search with the exception that not only are the scores for the highest output states used to generate a likelihood measure, but the algorithm also accumulates distance scores for states that should predict a low probability of state occupation. Figure 20 shows how durational models can be combined with the CDNTN subword models. In Figure 20, each subword model is comprised of a single CDNTN model trained to output the posterior probability of that subword occurring. The actual subword model is not restricted, and a discrete or continuous model can also be used. The structure of the model in Figure 20, is identical to that used for a hidden Markov model, except the transition probabilities in the hidden Markov model are replaced by explicit duration probability models. The duration of the subword is determined by comparing the state outputs to the duration model for that state using a dynamic programming algorithm. The state duration templates can be obtained directly from the subword durations in the training data set. A set of pre-derived state models can be used with the training utterance to generate a set of duration templates.

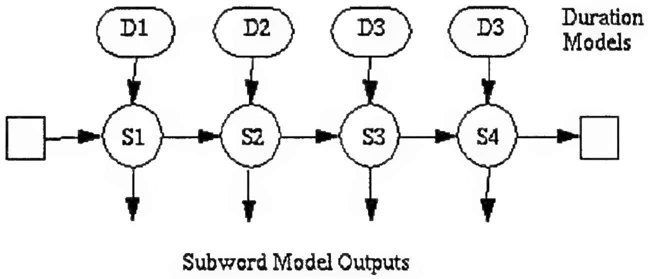


Figure 20: CDNTN Word Model

4.3.5 Nonparametric State Duration Modeling for Word Spotting

Conventional DTW word spotting systems typically use cepstral feature vectors for constructing the reference templates and test patterns [5]. An alternative to these meth- ods is to use the HMM state emission probabilities as template components in order to model the state durations. It is possible to construct a vector consisting of the state phone models within a keyword for each time unit as the reference template values. To create a reference template, a phoneme based HMM word model can be constructed in the manner outlined in the previous HMM based word spotting system. Once the state models have been parameterized, the feature vectors created from the keyword utterances used in the training data set can be passed through each phoneme model to generate a family of matri- ces consisting of state emission probabilities as a function of time. These templates can then be time aligned and averaged for all the keyword utterances in a similar manner to [25] to create a single reference template. This reference template contains the average state emission probabilities for each phoneme within the keyword as a function of time.

Using a DTW state duration model approach, minimizes the difference between a test template made from every state model at each time instant to the reference template. This distance is not only penalized by the most likely state having a low output probabil- ity, it also accounts for any state that has a different output than the template pattern. This method uses the additional information of certain states having low output emission prob- abilities at the same time as others having high outputs. In a phoneme based system, this corresponds to penalizing a word model that predicts multiple phonemes having high output probabilities at the same time, when the models should predict a single phonetic candidate. For a system using discriminatively trained state models, the use of all the state outputs maximizes the use of the training information where some states are parameter- ized to predict low probabilities for some feature vectors. This is in contrast to the Viterbi algorithm used in the HMM classifier where this information is lost by jumping to the state with the peak likelihood in the search path. Also, the conventional HMM classifier models the state durations by multiplying each duration output by connecting transition probabilities. The Viterbi algorithm uses the same basic method as described for the DTW system in 1.4, except that the test template is replaced by a simple time index, and can be considered as a subset of the more general dynamic programming algorithm. The best path is determined by the maximum log likelihood accumulated through the grid where each transition is multiplied by the appropriate transition probability.

4.4 The CDNTN Word Spotting System

A new word spotting system was developed using a CDNTN model for modeling phoneme output emission probabilities within an HMM framework. Subword durational probabilities were modeled using the nonparametric DTW template method described earlier.

4.4.1 Training the CDNTN Word Spotting System

As preliminary tests of the proposed word spotting system, initial state segmenta- tion of the phoneme boundaries of the keywords for the Road Rally speech Corpus was done using an HMM forced alignment. This was accomplished by training a serial network of three state HMM triphone models over all the utterances for each keyword. Once the HMM training was complete, the utterances were passed through the HMM networks and each feature vector was labeled according to the most likely phoneme model using a Viterbi search. This method was used to force a phonetically aligned segmentation for each keyword within the Waterloo section of the Road Rally Speech corpora. The forced alignment method can be replaced by one solely based on the CDNTN, and is described in [20]. A CDNTN segmental alignment procedure was implemented and exper- imental results showed a highly accurate segmentation is possible, when tested on a data base of phonetically segmented speech [20].

The Road Rally Speech Corpora was used to train and test the system. This data base consists of two independently recorded sections of different speakers from different dialects. The training section, known as the Waterloo Corpus is made up 56 speakers, 28 male and 28 female, reciting a paragraph about planning a road rally recorded through an actual telephone system sampled at 10KHz and filtered through a 300Hz to 3300Hz PCM FIR bandpass filter. The total duration of the Waterloo section consists of approximately two hours of read speech. Marking files for 20 keywords are provided, which specify loca- tions of the 20 keywords within the speech files. A separate test corpus called the Stonehenge section, is made up of free unrestricted conversational speech independently recorded between two speakers planning a road rally. The Stonehenge speech was recorded on high quality microphones, and filtered using a 300Hz to 3300Hz PCM FIR bandpass filter to simulate telephone bandwidth quality.

4.4.2 Growing CDNTN State Models

Once the speech data is phonetically segmented according to a phonetic dictionary, the phoneme segmentations were used to define subword states for discriminatively train- ing a CDNTN to predict the posterior probability of a subword given a training vector. The anticlass data used for each subword CDNTN model were the remaining subword vectors within the keyword labeled as not belonging to the subword being modeled. This amounts to training each subword model to predict the posterior output probability given a feature vector with respect to the other subwords within the keyword. CDNTN trees were grown for each phoneme occurring in each of the 20 keywords in the Road Rally Speech corpora. A total of 122 CDNTN trees were grown to model the subwords for the 20 keywords.

By restricting the training data used to develop the subword models to only phonetic data from that keyword, the CDNTN based keyword model can be used to define the most probable subword segmentations given a sequence putative keyword feature vectors. Alternate strategies for constructing the training data set have been tried, adding confused words and randomly selecting alternate keyword data as anticlass data. These methods were found to provide no substantial improvement in the keyword spotting system. A fundamental obstacle faced in using a discriminative classifier to construct subword models is that of defining an appropriate training set. A trade-off exists between trying to construct a global phonetic model using vast amounts of training data and one using locally appropriate data. Data sets using large amounts of anti-class data can mask the probability distribution of the specific phoneme model by artificially introducing a prior bias.

Once a training data set is established for each subword model, the CDNTN is grown using an L1 cost function. Pruning is necessary to allow a sufficient representation of each subword class for within each leaf region. The percentage based forward pruning method described in the appendix, combined with terminating growth at maximum level was found to provide the best results for the word spotting application. The incremental procedure of increasing the number of mixtures was used, using a hierarchical

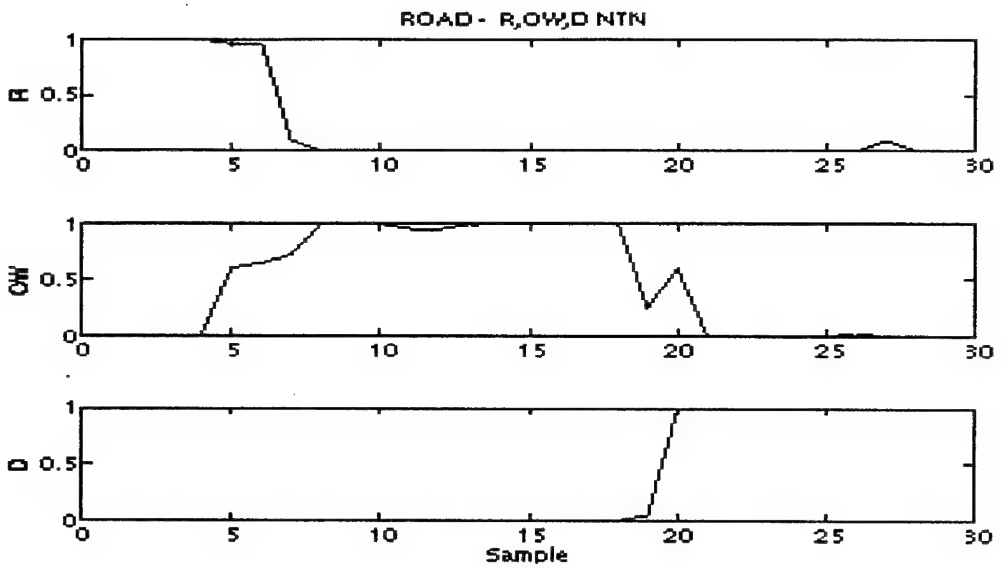


Figure 21: Sample CDNTN outputs for road word model

k-means algorithm.

4.4.3 Construction of Subword Duration Templates

Once the CDNTN subword models were created, temporally aligned feature vectors corresponding to each keyword utterance from the training data base were applied to each CDNTN phoneme model to obtain an subword state output vector corresponding to state emissions as a function of time for the keyword. Figure 21 shows a sample sequence of the CDNTN state models for an utterance of the keyword road by a speaker in the Waterloo section. Similar outputs were obtained for each keyword utterance for all keywords in the Waterloo section. Figure 21 illustrates how a single speaker dependent dura- tion template can be constructed from a single utterance of a keyword. To obtain a general speaker independent duration template, it is necessary to derive a template for each speaker in the training data base, and average the templates to form a general durational model.

A clustering method [25] was used to combine the duration templates generated from each keyword utterance in the Waterloo section using by a time aligned clustering method. The template clustering algorithm creates a template representing an average of individual templates time warped to the template with the minimum distance to every other template. The algorithm can be briefly described by constructing a two-dimensional grid, where each grid location i,j stores a distance $d(x_i, x_j)$ between each subword duration pattern x for each speaker. The distance metric $d(x_i, x_j)$, is defined as the distance obtained by dynamic time warping vector x_i with x_j, using a euclidean distance metric. Once the distance grid is computed, the duration vector which is found to have the smallest average distance to every other duration template is defined as the center template. Once the center template is found, each remaining duration vector is again time warped to align the sequences, and an average template is computed by averaging the time aligned duration outputs. Figure 22 shows the average template for the keyword road and Figure 23 shows another template for the keyword secondary. As can be seen in Figure 22 and Figure 23, the CDNTN provides a relatively good model for the subwords across all the Waterloo speakers. A poor subword model will result in a blurred average template. The duration template can be thought of as a matched filter for the subword outputs during a keyword. If a non keyword is presented to the system, the subword models will output low random probabilities which will in turn produce distortion values during

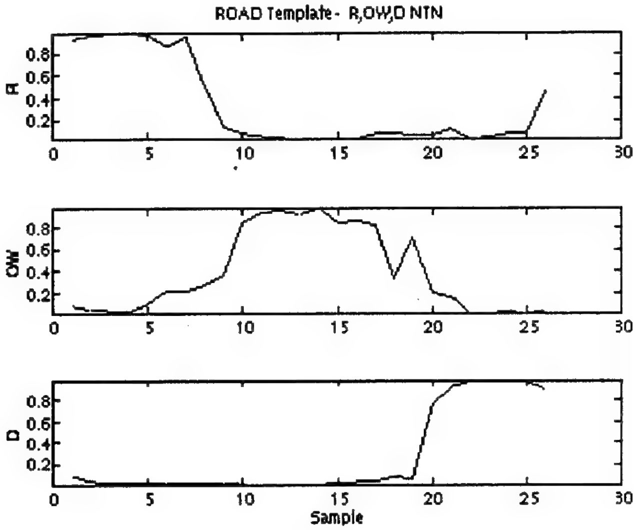


Figure 22: Averaged duration template for road state emissions

the template match.

To show the separation of distortion scores, an isolated word identification test was performed on a limited number of keywords in the Stonehenge section of the Road Rally Corpus. Figure 24 shows the distortion values for testing isolated utterances of retrace, conway, and road using the CDNTN duration template for road. A clear distinction can be seen in distortion levels obtained for road as compared to conway and retrace.

4.4.4 CDNTN Word Spotting System Description

To construct a word spotting system capable of real time performance, the DTW template matching scheme was performed in parallel for each word template by fixing a test template length based on the average template duration of each keyword. A sequence of output scores are generated by sliding this template along a stream of state emission probabilities generated from a continuous stream of speech data parameterized into 27 dimensional feature vectors and applied to the CDNTN model state generators. Figure 25 shows a diagram of the word spotting system for a single keyword. This new word spot- ting system differs from the previous HMM based system described earlier in that no background filler model is needed and putative keyword locations can be found indepen- dently without a network structure.

Construction of independent word spotting systems for each keyword allows the system to scale independently to the number of keywords, also allowing a simple parallelization of the system on a multi-processor network. A multi-word system can be made by running each independent keyword spotting system in parallel and combining the scores by a method which chooses the lowest score, shown in Figure 26.

Each output word score in Figure 25 is compared to a local keyword threshold to determine a putative keyword occurrence. A distance threshold can be extracted from histogram based on the scores of the keyword spotting system on a cross-validation data set for both correct hits and false alarms. The threshold location can then be found by choosing a value between the two distributions. An alternate method can use the results directly from the ROC curves for each keyword, to define the recognition level at a particular false alarm rate.

To obtain ROC scores a simple dynamic threshold was used. The dynamic running average of the output score was used as a threshold to output putative hit locations for construction of the ROC table.

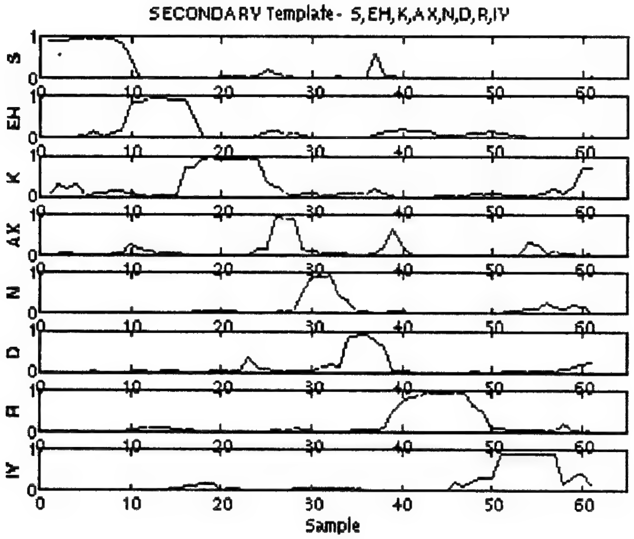


Figure 23: Averaged duration template for secondary state emissions

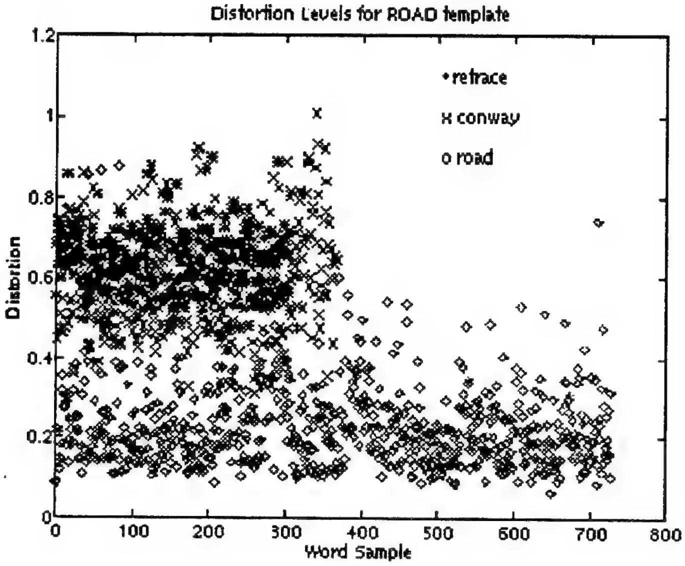


Figure 24: Distortion level for isolated keyword test for retrace, conway and road.

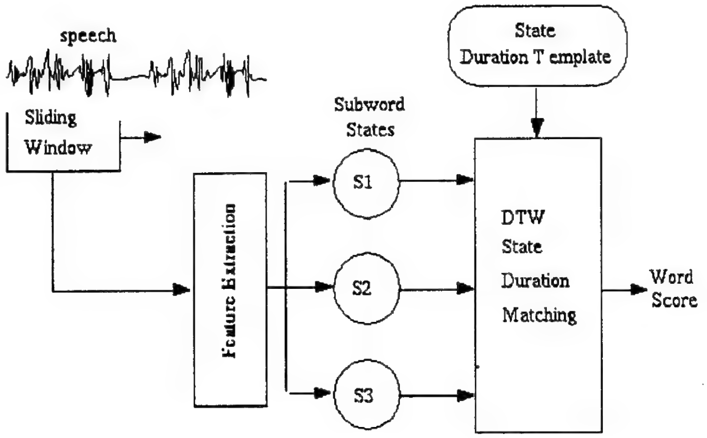


Figure 25: CDNTN word recognition system for a single word

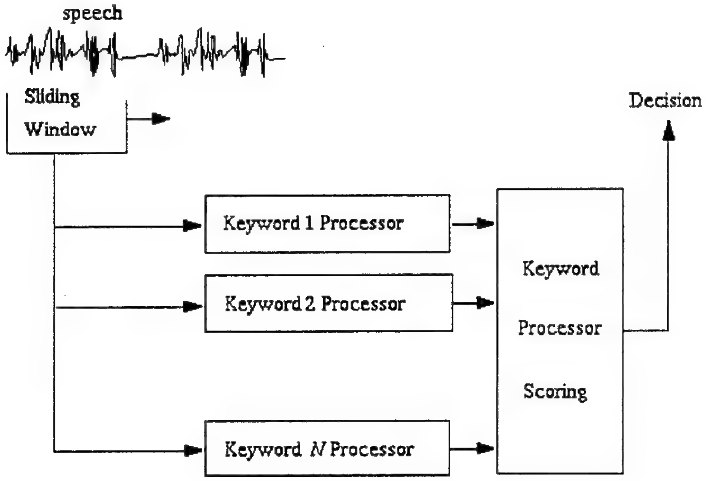


Figure 26: Multi word CDNTN word spotting system

The step size of the sliding template was set to 10duration template time for the entire keyword utterance. The short sliding window step size results in multiple low adjacent scores as the template passes through a keyword. Once the list of putative hits are obtained, the overlapping putative hits were merged to form a single hit. To allow calculation of a figure of merit, the sequence of putative hits obtained from each keyword was sorted into a list based on increasing distortion level. A figure of merit for recognition performance was used to evaluate the system for perfor- mance at 0 to 10 false alarm rates/keyword/hour in the same fashion described in. Since the CDNTN word spotting system uses a dynamic threshold, no attempt is made to limit the number of false alarms with low keyword output scores.

Figure 27 shows a sample output of the duration distortion for the keyword "spring- field" when the utterance "take the primary interstate west into Springfield" is spoken. The figure shows the distortion of the duration template algorithm as a function of feature sample. The keyword occurs in the utterance at the negative peak approximately at sample 325, and can be easily extracted by simply thresholding the output distortion.

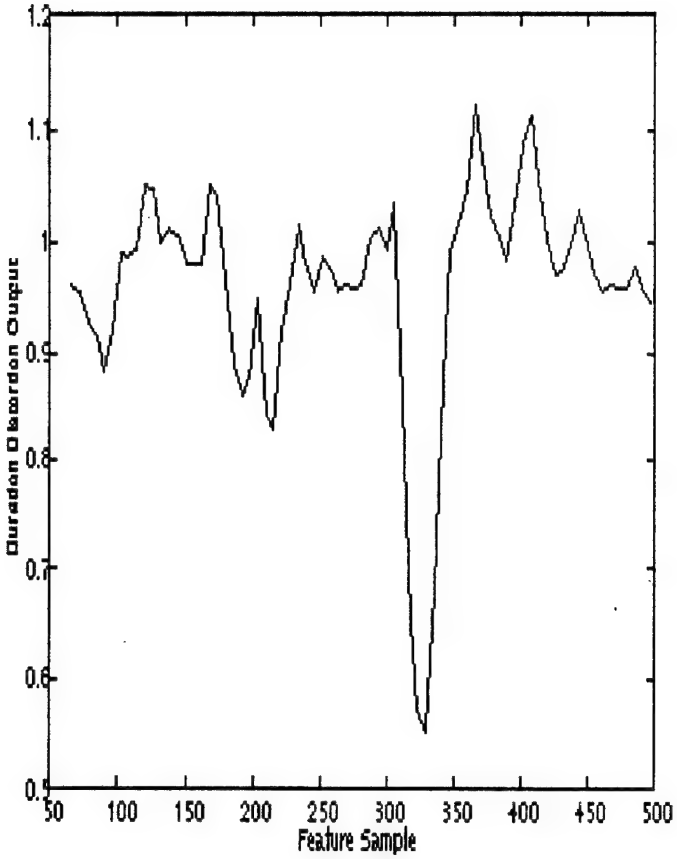


Figure 27: Sample output distortion of duration template match for “springfield”

5 Results

5.1 Results for Testing on 10 Male Speakers

The training algorithm for the NTN uses a sequential backpropagation update rule for training the perceptrons at each node. This means that the perceptron weights are updated after each feature vector. To prevent ordering effects of the data from producing a bias in the gradient decent, the training vectors are initially randomly ordered. Each new randomization allows a different local minimum to be found in the search space. To allow multiple minimums to be evaluated, the multiple weights are trained for each subword state model and tested against a cross-validation set to select the most optimum models. The NTN growing algorithm is also implemented with a mean splitting initialization of the perceptron to allow fast convergence. A minimum number of epochs for the percep- tron training was set a 200 to insure a minimum search time. A momentum term was added to the backpropagation algorithm to speed convergence of 0.4, for a step size of 0.5. Convergence was assumed when the difference of the mean square error measured every 10 epochs was less than 0.0001. During testing, the prior probabilities of each class was normalized to 0.5 to remove the bias created from the unequal amounts of anti-class data and in-class data, by the method described in the appendix.

Table 1 outlines the performance of the CDNTN model word spotting system for testing against 10 male speakers in the Stonehenge section of the Road Rally Speech Corpus. The features used were MFCC coefficients with added energy, and acceleration terms, with the mean removed. Each subword unit in this system was trained using a 5percentage based forward pruning method with a maximum tree level set to seven, as described in section appendix. A maximum number of mixtures created for both in-class and anti-class vectors in each leaf was limited to six. An iterative k-means clustering method was used, starting with a single mixture and incrementally increasing the number dynamically until less than n were assigned to each cluster. The limiting constant n was chosen as the dimension of the feature vector, in this case 27. The training data for this system used only the keyword utterances from each of the 56 speakers in the Waterloo section of the Stonehenge speech corpora. Each recited paragraph from the Waterloo section provided 99 keyword tokens for training per speaker. Discriminative training data for the subword units was selected as the feature vectors assigned to the remaining subwords within the keyword. Subsequently, each keyword model can be trained using only data from the utterances of that keyword from the training data base. The total amount of keyword tokens used for training this system was 5,544 for all 20 keywords. This averages to 277.2 tokens/keyword and 5 tokens/keyword/speaker.

Using the DTW scoring method with a fixed sliding window allows an actual distance metric to be defined for putative hits, thus a threshold can be set for each keyword according to Table 1. As can be seen in the Table 1, the performance varies between keywords. The total time of actual speech for the male test is approximately 44 minutes. Given this limited time, data error rates given in Table 1 are interpolated from the first 5 false alarms encountered in the ranked keyword lists.

each keyword, male only test

Figure 29 and Figure 30 show the histograms of the DTW distance scores for putative hits for retrace and secondary. As can be seen in the figures, the false alarm distortion measures are well above the majority of actual keyword scores. In these figures the keyword distortions were obtained directly from the DTW distance found between the test state duration template and the reference state duration template. For ROC curve estima- tion, the threshold was set dynamically as the instantaneous mean of the of the keyword score. This allowed a coarse threshold for simplified scoring. No attempt was made to limit the number of false alarms associated high distortion levels. This can be seen in Figure 29 and Figure 29 as high number of false alarms on the right side of the histograms.

Figure 32 shows the overall performance of the system for all 20 keywords up to an error rate of 10 FA's/Keyword/Hour for the male, female and cross-sex tests. As can be seen in the figure, the male

KeyWord	Percentage of Correct Hits allowing (x) False Alarms/Hour										
	0	1	2	3	4	5	6	7	8	9	10
BOONSBORO	33.8	37.7	41.5	45.3	47.1	47.1	47.4	48.6	49.9	50.0	50.0
CHESTER	7.6	14.2	20.7	27.3	30.3	30.3	30.9	33.6	36.2	36.4	36.4
CONWAY	44.4	44.4	44.4	44.4	48.0	54.4	59.3	59.3	59.3	59.3	59.3
INTERSTATE	15.0	15.0	15.0	15.0	23.3	38.5	50.0	50.0	50.0	54.1	58.4
LOOK	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MIDDLETON	59.1	59.1	59.1	59.1	59.1	59.1	59.1	59.1	59.1	59.1	59.1
MINUS	35.7	38.8	41.9	45.0	47.3	48.8	50.0	50.0	50.0	51.5	53.0
MOUNTAIN	9.5	10.6	11.8	13.0	13.5	13.5	13.5	13.5	13.5	13.5	13.5
PRIMARY	40.9	44.9	48.8	52.8	54.5	54.5	54.5	54.5	54.5	54.5	54.5
RETRACE	79.2	82.8	86.4	90.0	91.7	91.7	91.7	91.7	91.7	91.7	91.7
ROAD	7.3	7.3	7.3	7.3	7.9	9.0	9.8	9.8	9.8	9.8	9.8
SECONDARY	92.3	92.3	92.3	92.3	92.3	92.3	92.3	92.3	92.3	92.3	92.3
SHEFFIELD	66.7	66.7	66.7	66.7	66.7	66.7	66.7	66.7	66.7	66.7	66.7
SPRINGFIELD	71.4	71.4	71.4	71.4	72.6	74.6	76.7	78.8	80.8	81.0	81.0
THICKET	30.0	38.7	47.4	56.1	60.0	60.0	60.0	60.0	60.0	60.0	60.0
TRACK	15.0	16.4	17.9	19.3	20.8	22.2	23.3	23.3	23.3	23.3	23.3
WANT	0.0	0.0	4.1	8.9	11.1	11.1	11.1	11.1	11.1	11.1	11.1
WATERLOO	50.0	50.0	50.0	50.0	50.0	50.0	50.5	52.7	54.9	57.0	59.2
WESTCHESTER	61.8	64.3	66.9	69.4	70.6	70.6	70.6	70.6	70.6	70.6	70.6
BACKTRACK	0.0	0.0	36.8	80.3	100.	100.	100.	100.	100.	100.	100.
Overall	33.8	35.7	37.6	39.6	41.2	42.7	44.0	44.5	45.0	45.4	45.8

Figure 28: Word Spotting Performance for CDNTN trained phonetic subwords for each keyword, male only test

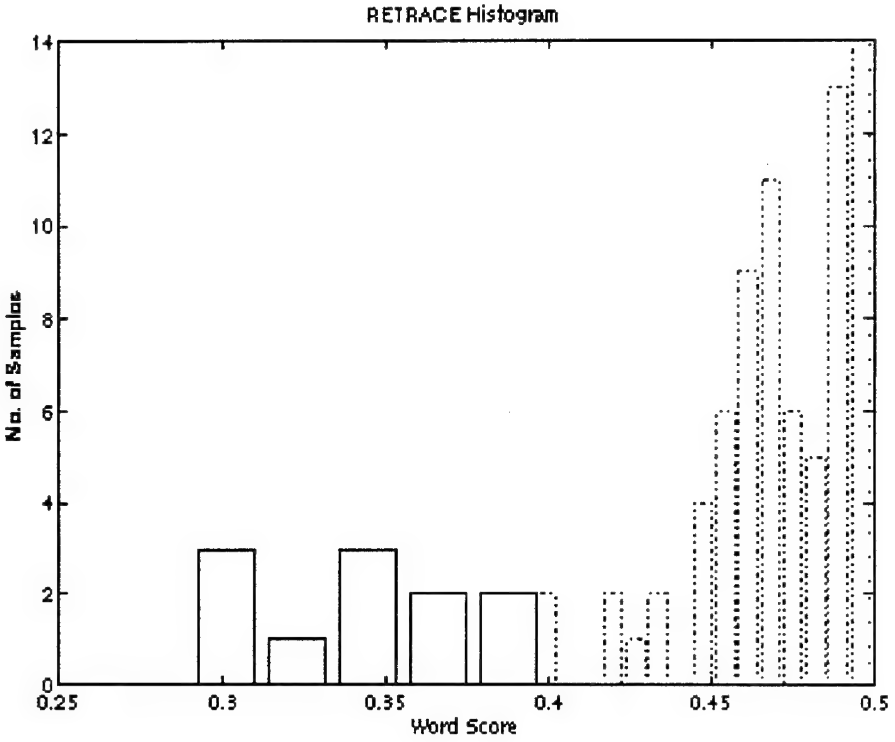


Figure 29: Histogram of keyword hits (solid) and false alarm (dotted) as a function of word score for retrace

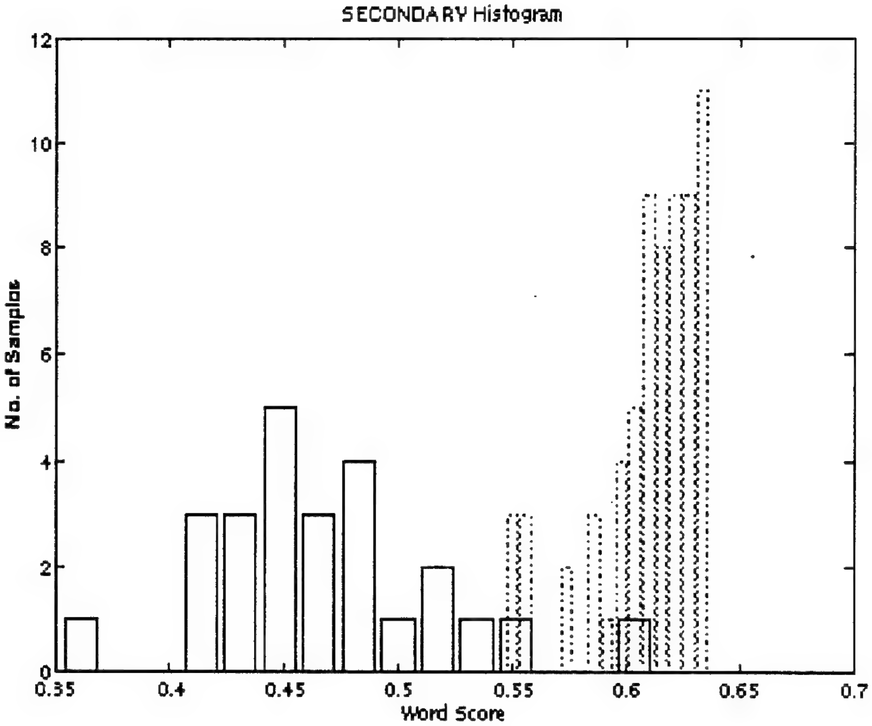


Figure 30: Histogram of keyword hits (solid) and false alarm (dotted) as a function of word scores for secondary

#	Test Conditions	#Hits	#FAs	#Actual	FOM	Hit Rate for 6 FA's/ hour
1	Cross Sex	750	24279	900	36.55%	39.0%
2	Male Test	360	13256	433	41.70%	44.0%
3	Female Test	390	11023	467	33.28%	35.5%
4	Male Test (16 Keywords)	285	8639	331	51.35%	53.9%

Figure 31: CDNTN Keyword Spotting Performance

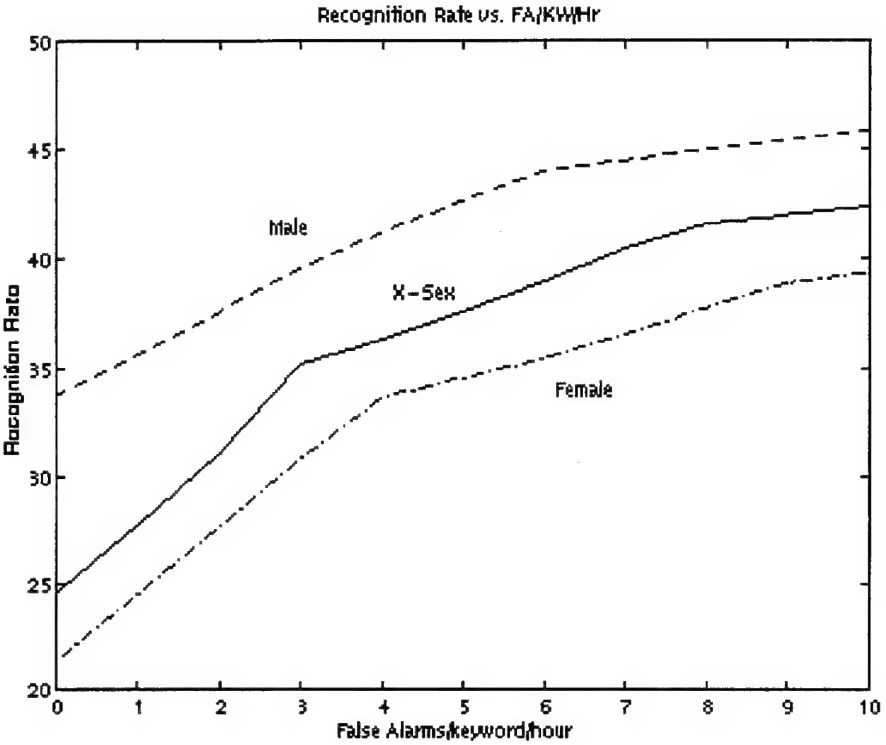


Figure 32: Overall performance for 20 keywords vs. FA/Keyword/Hour

speakers scored better than the female speakers. This is similar to the results found using the HMM word spotting system. Table 2 outlines the results for male, female and cross-sex tests. Since dynamic thresholding was used to obtain data for FOM scoring, no attempt was minimize false alarms at with low keyword scores. This is reflected in the total number of false alarms in the table. The high number of hits in Table 2 shows that this system can benefit greatly from a post processing method to further refine the keyword scores. A multi-level classifier system such as described by [11][12][28], has great potential for re-scoring false putative hits. ROC curves for each of the 20 keywords is shown in Figure 33 for male test. This figure shows that a large majority of the keywords performed well, while a subset of keywords brought down the average score. Figure 34 shows the average performance of the system for 16 multi-syllable keywords. Since no explicit background model is used, the shorter, simple keywords perform much worse that the longer keywords. This is a direct result from the fact that the longer keywords have more subword unit models which are more difficult to fit to random speech by the template matching duration model.

5.1.1 Performance as a Function of Training Parameters

A test was performed measuring the performance of the CDNTN word spotting system as a function of the maximum number of mixtures per class in the leaves. Table 3 gives the performance for the baseline system using a 5described in the previous section, with various maximum number of mixtures. The test was made on a cross validation set for 12 male speakers using conversational data. In general, both HMM and CDNTN systems performed worse on this test. Table 3 shows that increasing the maximum number of mixtures available, increases performance.

Table 4 gives the performance of the system as a function of the forward pruning threshold. The systems

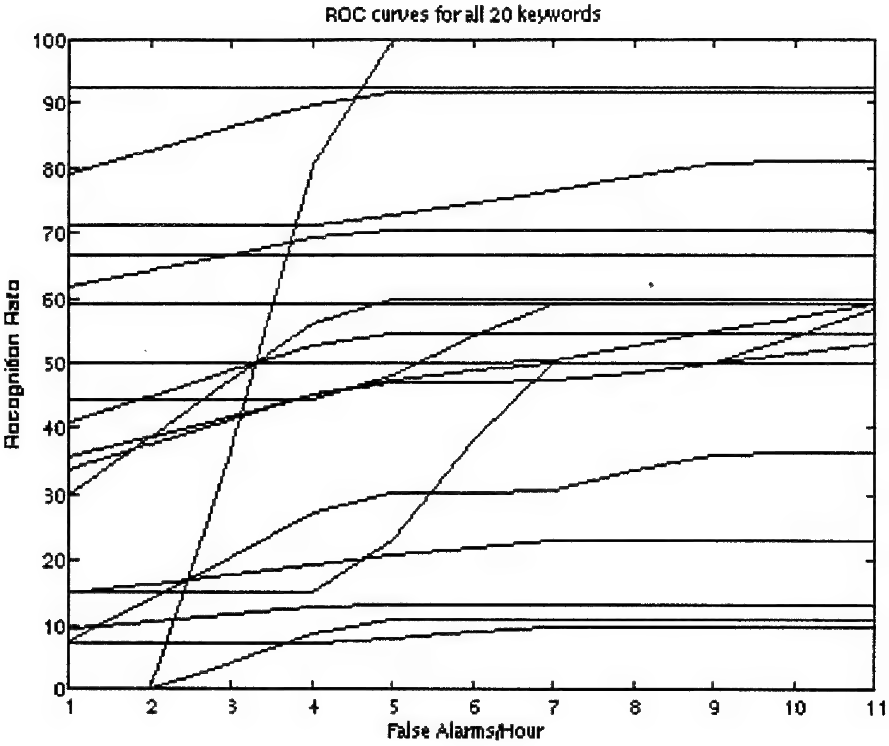


Figure 33: Individual performance for all 20 keywords as a function of FA/hour

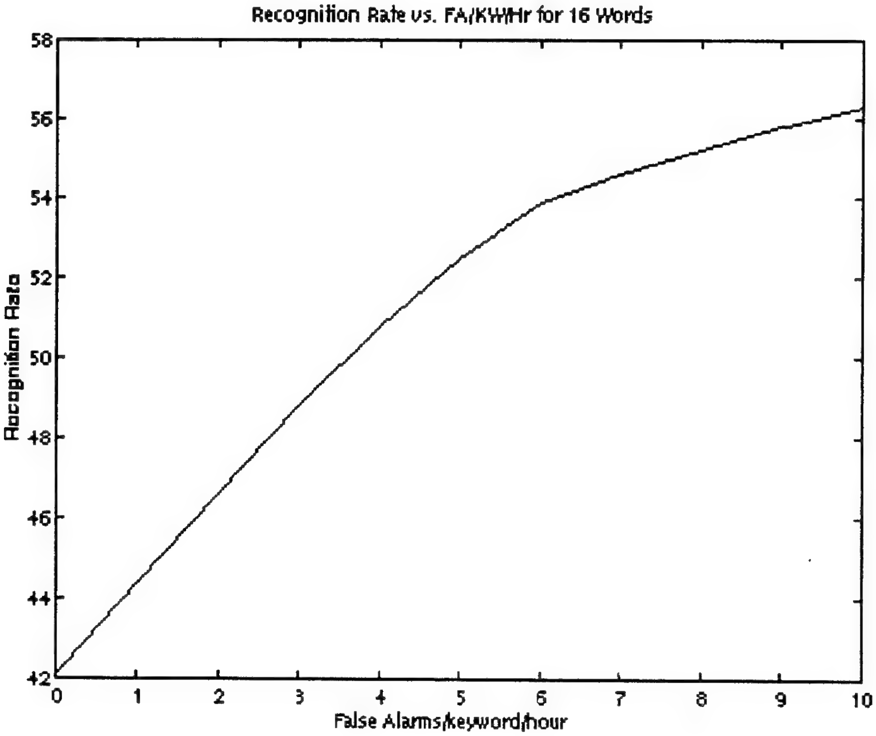
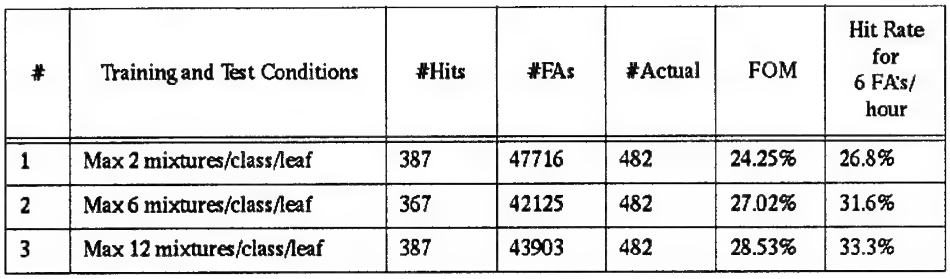
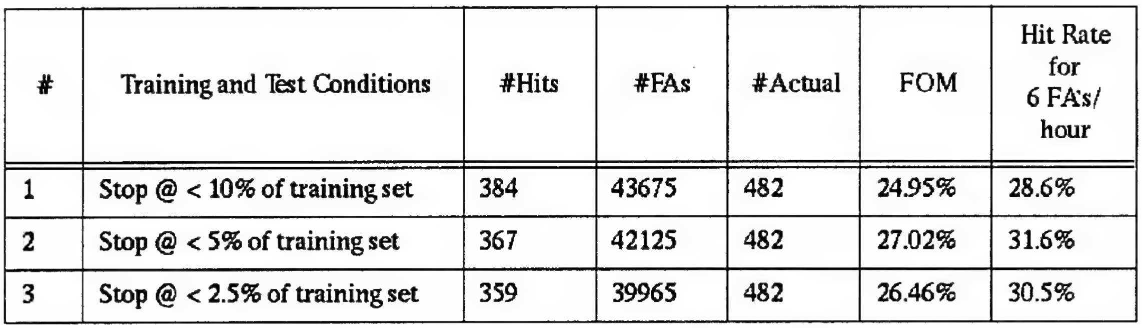


Figure 34: Performance for multi-syllable words (16 keywords)



#	Training and Test Conditions	#Hits	#FAs	#Actual	FOM	Hit Rate for 6 FA's/ hour
1	Max 2 mixtures/class/leaf	387	47716	482	24.25%	26.8%
2	Max 6 mixtures/class/leaf	367	42125	482	27.02%	31.6%
3	Max 12 mixtures/class/leaf	387	43903	482	28.53%	33.3%

Figure 35: CDNTN keyword spotting performance as a function of Gaussian mixtures on male cross validation set



#	Training and Test Conditions	#Hits	#FAs	#Actual	FOM	Hit Rate for 6 FA's/ hour
1	Stop @ < 10% of training set	384	43675	482	24.95%	28.6%
2	Stop @ < 5% of training set	367	42125	482	27.02%	31.6%
3	Stop @ < 2.5% of training set	359	39965	482	26.46%	30.5%

Figure 36: CDNTN keyword spotting performance as a function of forward pruning percentage level on male cross validation set

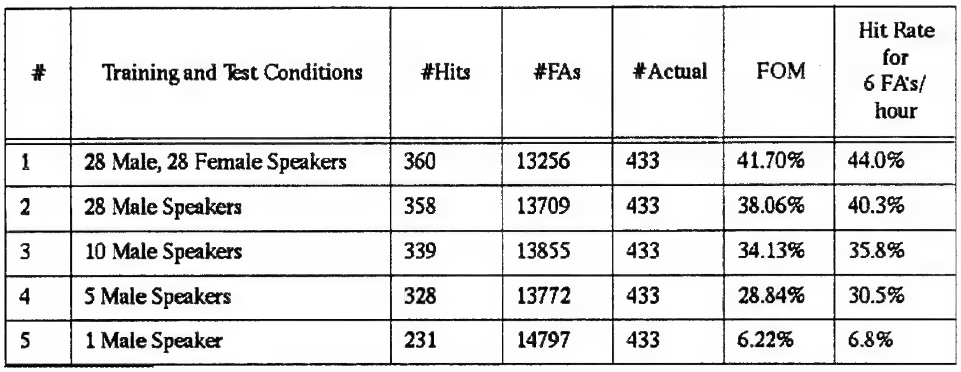
in the table were trained as described in the previous section, using a maximum of six mixtures/class/leaf. Table 4 shows that peak performance on the male cross validation set was obtained at 5.

5.1.2 CDNTN Word Spotting Performance with Reduced Data

One of the primary advantages of using a discriminate classifier for modeling the state occupations within a HMM model is the use of the additional data provided by the anti-class feature vectors. Discriminative training maximizes the use of costly training data. The CDNTN model provides an effective means for blending the attributes of both the continuous mixture model and the discriminative neural network. The grouping action of the NTN allows a efficient parametric model to be made for the vectors in each region of the features space. The separating hyperplanes defined by the perceptrons, partitions the feature space into high and low confidence regions. A minimal number of exemplars are necessary to define the regions defined by the NTN leaves.

To measure the performance of the CDNTN system as a function of training tokens, a number of systems were trained varying the number of speakers in the training set. Each system was trained using the identical training parameters of 5ing, seven maximum NTN levels, and six maximum mixtures/class/leaf. Table 5 gives the results for each system.

The recited paragraph used in the Waterloo section contains 99 keyword tokens, which amounts to approximately 5 tokens/keyword/speaker. Since no background model is assumed, no extra tokens are needed to train the system. This considerably reduces the cost involved in training the system in terms of providing marked speech files for training the system. For many applications such as monitoring keywords from a non cooperative subject, large amounts of speech data maybe impossible to obtain. The new word spotting system described maximizes the use of available tokens by the CDNTN state models to obtain superior



#	Training and Test Conditions	#Hits	#FAs	#Actual	FOM	Hit Rate for 6 FA's/ hour
1	28 Male, 28 Female Speakers	360	13256	433	41.70%	44.0%
2	28 Male Speakers	358	13709	433	38.06%	40.3%
3	10 Male Speakers	339	13855	433	34.13%	35.8%
4	5 Male Speakers	328	13772	433	28.84%	30.5%
5	1 Male Speaker	231	14797	433	6.22%	6.8%

Figure 37: CDNTN word spotting performance for varying amounts of training data for male test

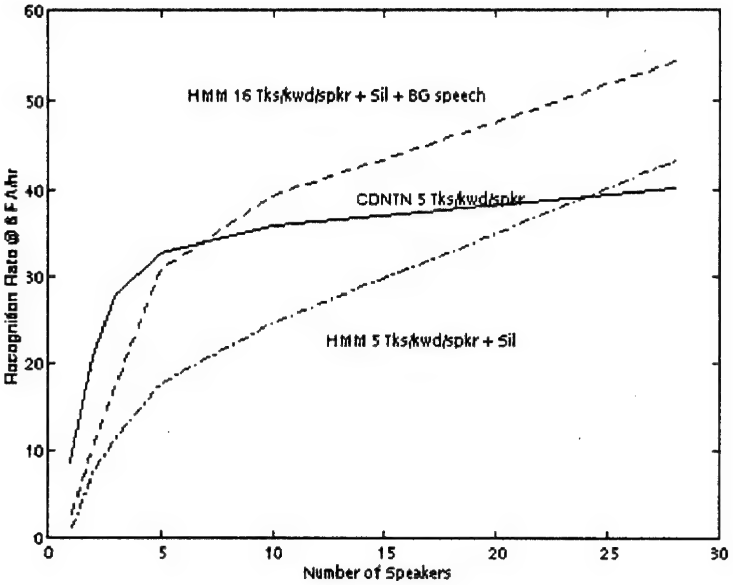


Figure 38: Comparison of CDNTN to HMM word spotting system with limited training speakers on male test

performance to comparable HMM systems. Figure 38 shows the system performance of the CDNTN word spotter compared to two HMM systems. The top HMM system using all the available training data from the Waterloo passage, contains a total of 321 tokens for training both the keywords and the background model not including the silence model. This translates to approximately 16 tokens/keyword/speaker. The CDNTN system requires only the keyword data, which translates to approximately 5 tokens/ keyword/speaker for training, with no extra data for modeling background silence. The cross over point between the best HMM system and the CDNTN system occurs between 6 and 7 speakers. Figure 38 also shows the performance of the HMM system trained using only pooled keyword data for both keyword triphone models and the background model, except for the silence model which uses background silence between keywords. A silence model is required for an HMM system to achieve non trivial performance. The compara- tively trained HMM system, shown in the dot-dashed line in Figure 39, performed worse than the CDNTN system when less than 25 speakers were used training. Figure 39 shows the performance for the different systems as a function of average number of training tokens used per keyword. The token counts do not include the tokens used to generate the silence models for the HMM systems.

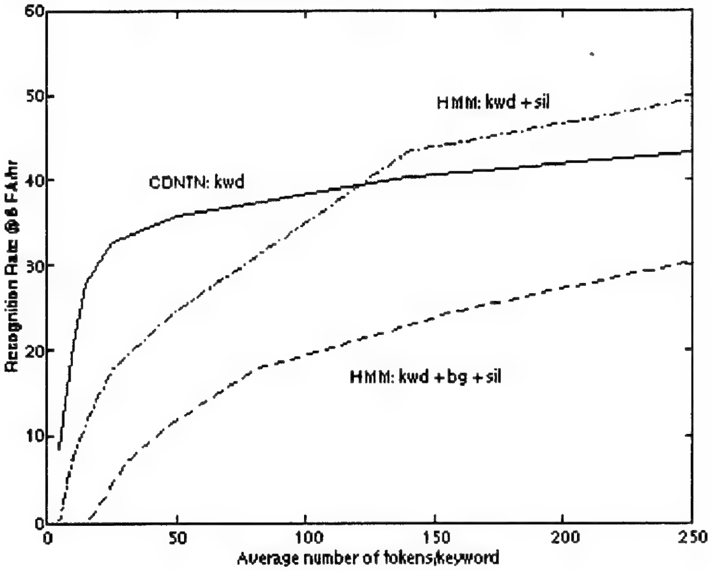


Figure 39: Comparison of CDNTN to HMM word spotting system as a function of training tokens on male test

References

[1] Bellman R, Dynamic Programming, Princeton University Press, Princeton, 1957.

[2] Bridle J.S, "An Efficient Elastic-Template Method for Detecting Given Words in Running Speech", British Acoustical Society Spring Meeting, Chelsea College, Paper No. 73SHC3, April 1973.

[3] Bridle J.S, "Pattern Recognition Techniques for Speech Recognition", J.C. Simon(ed.) Spoken Language Generation and Understanding, D. Reidel Publishing Co., pp. 129-145, 1980.

[4] Christiansen R.W, Rushforth C.K, "Detecting and Locating Key Words in Contin- uous Speech Using Linear Predictive Coding", IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-25, No. 5, pp. 361-367, Oct. 1977.

[5] Deller J.R.Jr, Proakis J.G, Hansen J.H.L, Descrete-Time Processing of Speech Sig-nals, Macmillan Publishing Co., New York, 1993.

[6] Higgins A.L, Wohlford R.E, "Keyword recognition using template concatenation", Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 1233-1236, April 1985.

[7] Hwang J-N, Vlontzos J.A, Kung S-Y, "A Systolic Neural Network Archecture for Hidden Markov Models", IEEE Trans. on Acoustics, Speech and Signal Proc., Vol. 37, No. 12, Dec. 1989.

[8] Jacobs R.A, Jordan M.I, Nowlan S.J, Hinton G.E, "Adaptive Mixtures of Local Experts", Neural Computation, Vol. 3, pp. 79-87, 1991.

[9] Jacobs R.A, Jordan M.I, "Learning Piecewise Control Strategies in a Modular Neural Network Architecture", IEEE Trans. on Systems, Man and Cybernetics, Vol. 23. No. 2, March/April 1993.

[10] Li K.P, Naylor J.A, "Whole Word Recurrent Neural Network for Keyword Spot- ting", Int. Conf. on Acoustics, Speech and Signal Processing, Vol. II, pp.81-84, 1992.

[11] Morgan D.P, Scofield C.L, Lorenzo T.M, Real E.C, Loconto D.P, "A Keyword Spotter Which Incorporates Neural Netoworks for Secondary Processing", Int. Conf. on Acoustics, Speech and Signal Processing, New Mexico, Vol. 1, pp. 113- 116, 1990.

[12] Morgan D.P, Scofield C.L, Adcock J.E, "Multiple Neural Network Topologies Applied to Keyword Spotting", International Conf. on Acoustics, Speech and Sig- nal Proc., pp. 313-316, 1991.

[13] Myers C.S, Rabiner L.R, Rosenberg A.E, "An Investigation of the Use of Dynamic Time Warping for Word Spotting and Connected Speech Recognition", Int. Conf. on Acoustics, Speech and Signal Processing, Denver, pp. 173-177, 1980.

[14] Paul D.B, "Speech Recognition Using Hidden Markov Models", The Lincoln Lab- oratory Journal, Vol. 3. No. 1, 1990.

[15] Peinado A.M, et.al. "Improvements in HMM-based isolated word recognition sys- tem", IEE Proceedings-I, Vol 138. No. 3, pp. 201-206, June 1991.

[16] Rabiner L, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE, vol. 77, no. 2, Feb. 1989.

[17] Rabiner L, Juang B-H, Fundamentals of Speech Recognition, Prentice Hall, Engle- wood Cliffs, 1993.

[18] Rose R.C, Paul D.B, "A Hidden Markov Model Based Keyword Recognition Sys- tem", Intr. Conf. on Acoustics Speech and Signal Processing, vol I, pp. 129-132, 1990.

[19] Sankar A, Mammone R.J, "Speaker Independent Vowel Recognition using Neural Tree Networks", Proc. of the Int. Joint Conf. on Neural Networks, Seattle, 1991

[20] Sharma M, Mammone R.J, "Speech Recognition Using Sub-Word Neural Tree Network Models and Multiple Classifier Fusion", submmitted to the 1995 Int. Conf. on Acoustics, Speech and Signal Processing.

[21] Sivakumar S.C, Robertson W, Macleod K, "Improving Temporal Representation in TDNN Structure For Phoneme Recognition", Proc. the Int. Joint Conf. on Neural Networks, Baltimore, June 1992.

[22] Spohrer J.C, Brown P.F, Hochschild P.H, Baker J.K, "Partial backtrace in continuous speech recognition", Proc. Int. Conf. on Systems, Man, and Cybern etics, pp. 36-42, 1980.

[23] Vicenzi C. et.al., "Large vocabulary isolated word recognition: a real time implementation", IEE Proceedings I, Communications, Speech and Vision, Vol. 136. No. 2, pp. 127- 132, April, 1989.

[24] Wilpon J.G, Rabiner L.R, "On the Recognition of Isolated Digits From a Large Telephone Customer Population", The Bell System Technical Journal, Vol. 62, No. 7, pp. 1977-2000, Sept. 1983.

[25] Wilpon J.G, Rabiner L.R, "A Modified K-Means Clustering Algorithm for Use in Isolated Word Recognition", IEEE Trans. on Acoustics, Speech, and Signal Pro- cessing, Vol. ASSP-33, June, 1985.

[26] Wilpon J.G, Rabiner L.R, Lee C-H, Goldman E.R, "Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models", IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. 38. No. 11, Nov. 1990.

[27] Wilpon J.G, DeMarco D.M, Mikkilinei R.P, "Isolated word recognition over the DDD telephone network - results of two extensive field studies", Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, 1S.1.10, pp. 55-57, Apr. 19 1988.

[28] Wu J, Chan C, "Isolated Word Recognition by Neural Network Models with Cross-Correlation Coefficients for Speech Dynamics", IEEE Transactions on Pat- tern Analysis and Machine Intelligence, Vol. 15, No. 11, pp. 1174-1185, Nov. 1993.

[29] Zeppenfeld T, Waibel A.H, "A Hybrid Neural Network, Dynamic Programming Word Spotter", International Conf. on Acoustics, Speech and Signal Proc., Vol II. pp. 77-80, 1992.

[30] A. S. Atal. Speech Analysis/Synthesis by Linear Prediction of the Speech wave. *J. Acoust. Soc. Am.*50, pp. 637-655, 1971.

[31] J. Makhoul Linear Prediction: A Tutorial Review. *proc. IEEE,* 63, pp. 561-580, 1975.

[32] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition.* Prentice Hall, Englewood Cliffs, NJ, 1993.

[33] D. Reynolds. Evaluation of different features for speaker identification. *Presentation at the Robust Speech Recognition Workshop*, Rutgers University, August 1993.

[34] Yu-Hung Kao *Robustness Study of Free-Text Speaker Identification and Verification* PhD thesis, The University of Maryland, December, 1992.

[35] B. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *Journal of the Acoustical Society of America*, 55:1304–1312, June 1974.

[36] S. Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Trans. Acoust., Speech, Signal Process.*, ASSP-29:254–272, April 1981.

[37] J.P. Campbell. *Features and measures for speaker recognition.* PhD thesis, Oklahoma State University, December 1992.

[38] F.K. Soong, A.E. Rosenberg, L.R. Rabiner, and B.H. Juang. A vector quantization approach to speaker recognition. In Proc. Int. Conf. Acoust., Speech, Signal Process., pages 387–390, 1985.

[39] G. Velius. Variants of cepstrum based speaker identity verification. In *Proc. Int. Conf. Acoust., Speech, Signal Process.*, pages 583–586, 1988.

[40] M. R. Schroeder. Direct (nonrecursive) relations between cepstrum and and predictor c oefficients. *IEEE Trans. Acoust. Speech, Signal Process.*, **29**:297–301, Apr. 1981.

[41] B.H. Juang, L. R. Rabiner, and J. G. Wilpon. On the use of bandpass liftering in speech recognition, *IEEE Trans. Acoust., Speech, Signal Process.* **ASSP-35** (7): 947-954, July 1987.

[42] K. K. Paliwal. On the performance of the quefrency weighted cepstral coefficients in vowel recognition. *Speech Commun.*, **1**:151-154,1982.

[43] Y. Tohkura. A weighted cepstral measure for speech recognition. *IEEE Trans. Acoust., Speech, Signal Process.* **ASSP-35** (10): 947-954, Oct. 1987.

[44] Spectral slope based distortion measure for all pole models of speech. *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 757-760 (1986)

[45] F. K. Soong and A. E. Rosenberg. On the use of instantaneous and transitional spectral information in speaker recognition. *IEEE Trans. Acoust., Speech, Signal Process.*, ASSP-36:871–879, June 1988.

[46] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn. RASTA-PLP Speech Analysis Technique. *Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing,* San Francisco, 1992, I-121–124.

[47] S. Furui. On the role of dynamic characterstics of speech spectra for syllable perception. *Fall Meeting of Acoust. Soc. Japan*, 1-1-2: Oct. 1984.

[48] F. K. Soong, A. E. Rosenberg, L. R. Rabiner, and B. H. Juang. A vector quantization approach to speaker recognition. In *Proceedings ICASSP*, pages 387–390, 1985.

[49] K. R. Farrell, R. J. Mammone, and K. T. Assaleh. Speaker recognition using neural networks and conventional classifiers. *IEEE Trans. on Speech and Audio Processing*, 2(1), part 2, 1994.

[50] L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their applications to hand-written character recognition. *IEEE Trans. on Systems, Man and Cybernetics*, 23(3):418–435, 1992.

[51] Henderson and Weitz. Multisensor knowledge systems. *International Journal of Robotics Research*, 7(6):114–137, June 1988.

[52] J. A. Benediktsson and P. H. Swain. Consensus theoretic classification methods. *IEEE Trans. on Systems, Man and Cybernetics*, 22(4):688–704, 1992.

[53] T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(1):66–75, 1994.

[54] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Trans. on Communications*, COM-28:84–95, 1981.

[55] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, NY, 1973.

[56] A. Sankar and R. J. Mammone. Growing and pruning neural tree networks. *IEEE Trans. on Computers*, C-42:221–229, March 1993.

[57] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth International Group, Belmont, CA, 1984.

[58] R. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, pages 4–22, Apr. 1987.

[59] S. Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-29:254–272, April 1981.

[60] B. Atal. Automatic recognition of speakers from their voices. *Proceedings of the IEEE*, 64:460–475, April 1976.

[61] A. Rosenberg. Automatic speaker recognition: A review. *Proceedings of the IEEE*, 64:475–487, April 1976.

[62] G. Doddington. Speaker recognition - identifying people by their voices. *Proceedings of the IEEE*, 73:1651–1664, 1985.

[63] J.L. Flanagan. *Speech Analysis, Synthesis, and Production.* Springer-Verlag, 1972.

[64] S. Furui and F. Itakura. Talker recognition by statistical features of speech sounds. *Electronic Communication*, 56-A:62–71, 1973.

[65] A.V. Oppenheim and R.W. Schafer. Homomorphic analysis of speech. *IEEE Transactions on Audio and Electroacoustics*, AU-16:221–226, June 1968.

[66] D. O'Shaughnessy, Speech communication, human and machine, Addison-Wesley series in Electrical Engineering: Digital Signal Processing.

[67] K. T. Assaleh and R. J. Mammone, "New LP-derived features for speaker identification", *IEEE Trans. on Speech and Audio Proc.*, vol. 2, pp. 630–638, Oct. 1994.

[68] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, 1973.

[69] G. Arfken, *Mathematical Methods for Physicists*, Academic Press, 1985.

[70] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, 1992.

MISSION
OF
ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Materiel Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.